TRACKING OF CLASS ATTENDANCE USING RFID AND THUMBPRINT IDENTIFICATION DEVICE

ANITH SAFURA BT AZMI

ELECTRICAL & ELECTRONICS ENGINEERING
UNIVERSITI TEKNOLOGI PETRONAS
JUNE 2010

TRACKING OF CLASS ATTENDANCE USING RFID SYSTEM INTEGRATED WITH THUMBPRINT IDENTIFICATION TECHNOLOGY

by

ANITH SAFURA BINTI AZMI

FINAL PROJECT REPORT

Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

© Copyright 2010
by
Anith Safura Binti Azmi, 2010

CERTIFICATION OF APPROVAL

TRACKING OF CLASS ATTENDANCE USING RFID SYSTEM INTEGRATED WITH THUMBPRINT IDENTIFICATION TECHNOLOGY

by

Anith Safura Binti Azmi

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

Puan Hanita Binti Daud
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

June 2010

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

ABSTRACT

The objective of this project is to use the combination of Radio-Frequency Identification and thumbprint scanner as tracking tool for class attendance to replace the old attendance taking that can easily been cheated by the students. RFID is the use of an object (typically referred to as an RFID tag) applied to or incorporated into a product, animal, or person for the purpose of identification and tracking using radio waves. Some tags can be read from several meters away and beyond the line of sight of the reader. This project will use this device to make sure the students that come to the class are attending to the class. As for thumbprint detector, it will act as an attendance collector since we can not copy or steal someone identification using it and send the data about the owner's thumbprint. This project combines these two systems and helps the lecturers to face the dishonesty problem on class attendance. Since student always helps his or her friend by signing the class attendance for his or her friend, therefore I am proposing this project that detects the attendances of the real person.

ACKNOWLEDGEMENTS

Firstly, I would like to thank the Almighty God for giving me the strength and time to complete this project.

To both of my parents and my family, thank you for your love and support while I am facing the hardship in the project.

I also would like to give my most gratitude to my supervisor, Mdm Hanita Daud for giving me support and was always understanding despite all the errors and delays that I made during the completion of the project.

I also would like to give a thousand thanks to the Mr. Shahjehan Tahir for his cooperation, knowledge sharing, endless guidance and advice on the work in order to complete the project successfully.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Background of Study

Tracking of class attendance using Radio Frequency Identification (RFID) is one of the newest technologies that have been applied in some universities in some countries. This system can avoid the students from escaping the lectures. It is also can lighten the burden for the lecturer to key in the record of the attendances. The RFID systems will make sure that the students are being in the class during the lecture hours. Some advantages that these systems give are it decreases the loss time for the students and lecturers and the lecturers can know exactly the students who attended a particular class. As for thumbprint scanner, this technology has been used in many companies as attendances since thumbprint is a unique identification for every person in this world. The thumbprint time attendance system supports the electronic storage of data. The captured thumbprint patterns gets directly stored in the database of the computer. Thus, data loss is less. Not like the earliest method where data used to be maintained in the paper register and missing of just a page used to cause severe loss

1.2 Problem Statement

Nowadays, students have many different ways to cheat on their class attendance. Some students will ask their friend helps to sign for them when ever they feel like escaping the class. Although the lecturer tries to vary the style of taking the class attendance by calling the student's name but the students will be one step forward by asking for their friend to answer for them. Hence, the new technology needs to be used to avoid these students from continuously cheating in their class attendance.

1.3 Objectives and Scope of study

The objective of the project is to develop a system that can detect the presence of the students during class. This system will make sure the student is being in the class during the lecture time. The RFID is being used as a checker to track the student during class and will be captured in the database while the thumbprint scanner will be used as an attendance. In order to achieve the objectives of this project, some research and tasks need to be carried out by collecting all the technical details and programming regarding the RFID and thumbprint scanner. The programming for both systems is also needed to be focused and understood to make them become one new system. A recommendation is to be made by the findings of the applicability of the RFID and thumbprint scanner.

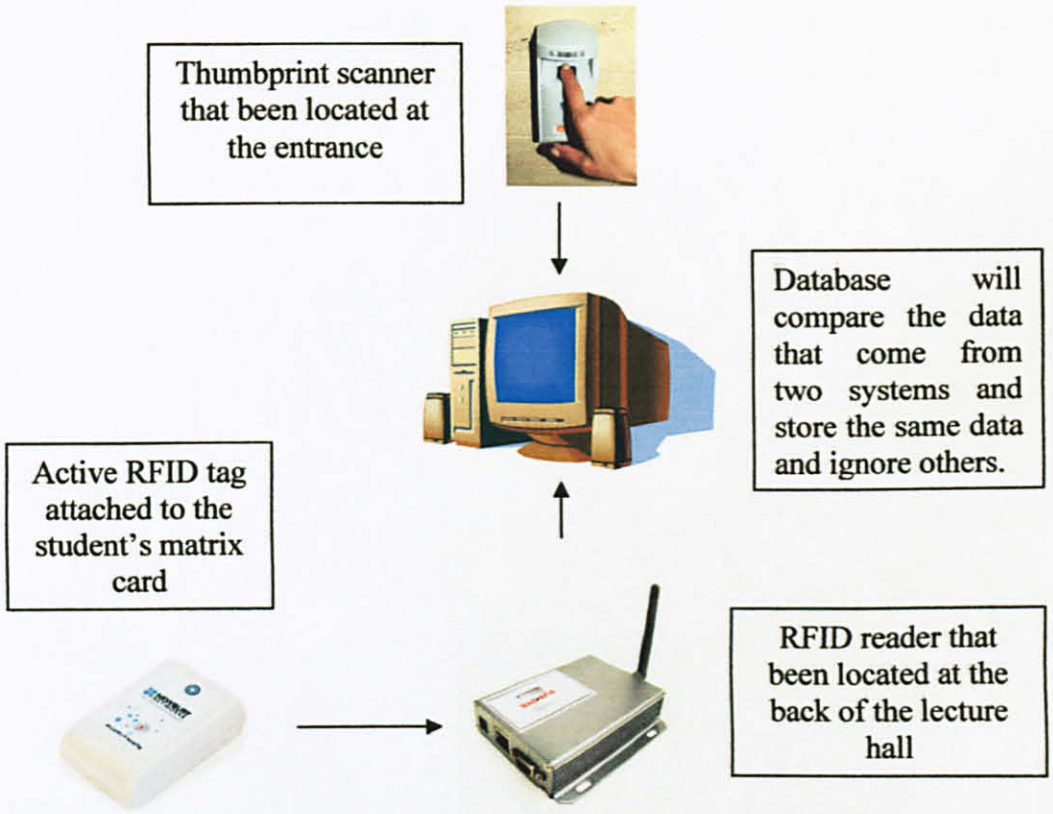

Figure 1: Block diagram on how the system runs

CHAPTER 2
LITERATURE REVIEWS

2.1 Thumbprint Basic

Thumbprint is a one unique identification for every person in this world. We can not copy or steal someone identification using thumbprint. The thumbprint scanner is working as same like the fingerprint scanner works. A fingerprint is made up of a pattern of ridges and furrows as well as characteristics that occur at Minutiae points (ridge bifurcation or a ridge ending). Fingerprint scanning essentially provides an identification of a person based on the acquisition and recognition of those unique patterns and ridges in a fingerprint [12].

Figure 2: Thumbprint

People have tiny ridges of skin on their fingers because this particular adaptation was extremely advantageous to the ancestors of the human species. The pattern of ridges and "valleys" on fingers make it easier for the hands to grip things,

in the same way a rubber tread pattern helps a tire grip the road. The other function of fingerprints is a total coincidence. Like everything in the human body, these ridges form through a combination of genetic and environmental factors. The genetic code in DNA gives general orders on the way skin should form in a developing fetus, but the specific way it forms is a result of random events. The exact position of the fetus in the womb at a particular moment and the exact composition and density of surrounding amniotic fluid decides how every individual ridge will form [13].

So, in addition to the countless things that go into deciding your genetic make-up in the first place, there are innumerable environmental factors influencing the formation of the fingers. Just like the weather conditions that form clouds or the coastline of a beach, the entire development process is so chaotic that, in the entire course of human history, there is virtually no chance of the same exact pattern forming twice.

Consequently, fingerprints are a unique marker for a person, even an identical twin. And while two prints may look basically the same at a glance, a trained investigator or an advanced piece of software can pick out clear, defined differences. This is the basic idea of fingerprint analysis, in both crime investigation and security. A fingerprint scanner's job is to take the place of a human analyst by collecting a print sample and comparing it to other samples on record

2.2 Thumbprint Sensor

A fingerprint sensor is an electronic device used to capture a digital image of the fingerprint pattern. The captured image is called a live scan. This live scan is digitally processed to create a biometric template (a collection of extracted features) which is stored and used for matching. This is an overview of some of the more commonly used fingerprint sensor technologies.

2.2.1 Optical

Optical fingerprint imaging involves capturing a digital image of the print using visible light. This type of sensor is, in essence, a specialized digital camera. The top layer of the sensor, where the finger is placed, is known as the touch surface. Beneath this layer is a light-emitting phosphor layer which illuminates the surface of the finger. The light reflected from the finger passes through the phosphor layer to an array of solid state pixels (a charge-coupled device) which captures a visual image of the fingerprint. A scratched or dirty touch surface can cause a bad image of the fingerprint. A disadvantage of this type of sensor is the fact that the imaging capabilities are affected by the quality of skin on the finger. For instance, a dirty or marked finger is difficult to image properly. Also, it is possible for an individual to erode the outer layer of skin on the fingertips to the point where the fingerprint is no longer visible. It can also be easily fooled by an image of a fingerprint if not coupled with a "live finger" detector. However, unlike capacitive sensors, this sensor technology is not susceptible to electrostatic discharge damage [14]

2.2.2 Ultrasonic

Ultrasonic sensors make use of the principles of medical ultrasonography in order to create visual images of the fingerprint. Unlike optical imaging, ultrasonic sensors use very high frequency sound waves to penetrate the epidermal layer of skin. The sound waves are generated using piezoelectric transducers and reflected energy is also measured using piezoelectric materials. Since the dermal skin layer exhibits the same characteristic pattern of the fingerprint, the reflected wave measurements can be used to form an image of the fingerprint. This eliminates the need for clean, undamaged epidermal skin and a clean sensing surface [14].

2.2.3 Capacitance

Capacitance sensors utilize the principles associated with capacitance in order to form fingerprint images. The two equations used in this type of imaging are:

$$C = \frac{Q}{V}$$

$$C = \varepsilon_0 \varepsilon_r \frac{A}{d}$$

where

C is the capacitance in farads
Q is the charge in coulombs
V is the potential in volts
ε_0 is the permittivity of free space, measured in farads per metre
ε_r is the dielectric constant of the insulator used
A is the area of each plane electrode, measured in square metres
d is the separation between the electrodes, measured in metres

In this method of imaging, the sensor array pixels each act as one plate of a parallel-plate capacitor, the dermal layer (which is electrically conductive) acts as the other plate, and the non-conductive epidermal layer acts as a dielectric [14].

2.2.4 Passive capacitance

A passive capacitance sensor uses the principle outlined above to form an image of the fingerprint patterns on the dermal layer of skin. Each sensor pixel is used to measure the capacitance at that point of the array. The capacitance varies between the ridges and valleys of the fingerprint due to the fact that the volume between the dermal layer and sensing element in valleys contains an air gap. The dielectric constant of the epidermis and the area of the sensing element are known values. The measured capacitance values are then used to distinguish between fingerprint ridges and valleys [14]

2.2.5 Active capacitance

Active capacitance sensors use a charging cycle to apply a voltage to the skin before measurement takes place. The application of voltage charges the effective capacitor. The electric field between the finger and sensor follows the pattern of the ridges in the dermal skin layer. On the discharge cycle, the voltage across the dermal layer and sensing element is compared against a reference voltage in order to calculate the capacitance. The distance values are then calculated mathematically, using the above equations, and used to form an image of the fingerprint [13]. Active capacitance sensors measure the ridge patterns of the dermal layer like the ultrasonic method. Again, this eliminates the need for clean, undamaged epidermal skin and a clean sensing surface [14].

2.3 Radio-Frequency Identification (RFID)

Radio-frequency identification (RFID) is the use of an object applied to or incorporated into a product, animal, or person for the purpose of identification and tracking using radio waves. Some tags can be read from several meters away and beyond the line of sight of the reader. Radio-frequency identification comprises interrogators (also known as readers), and tags (also known as labels). Most RFID tags contain at least two parts. One is an integrated circuit for storing and processing information, modulating and demodulating a radio-frequency (RF) signal, and other specialized functions. The second is an antenna for receiving and transmitting the signal [5].

RFID systems can be used just about anywhere, from clothing tags to missiles to pet tags to food -- anywhere that a unique identification system is needed. The tag can carry information as simple as a pet owners name and address or the cleaning instruction on a sweater to as complex as instructions on how to assemble a car. Some auto manufacturers use RFID systems to move cars through an assembly line. At each successive stage of production, the RFID tag tells the computers what the next step of automated assembly is [7].

One of the key differences between RFID and bar code technology is RFID eliminates the need for line-of-sight reading that bar coding depends on. Also, RFID scanning can be done at greater distances than bar code scanning. High frequency RFID systems (850 MHz to 950 MHz and 2.4 GHz to 2.5 GHz) offer transmission ranges of more than 90 feet, although wavelengths in the 2.4 GHz range are absorbed by water (the human body) and therefore has limitations [7].

There are generally three types of RFID tags: active RFID tags, which contain a battery and can transmit signals autonomously, passive RFID tags, which have no battery and require an external source to provoke signal transmission, and battery assisted passive (BAP) RFID tags, which require an external source to wake up but have significant higher forward link capability providing greater range.

2.4 Active RFID Tag

An RFID tag is an active tag when it is equipped with a battery that can be used as a partial or complete source of power for the tag's circuitry and antenna [5].

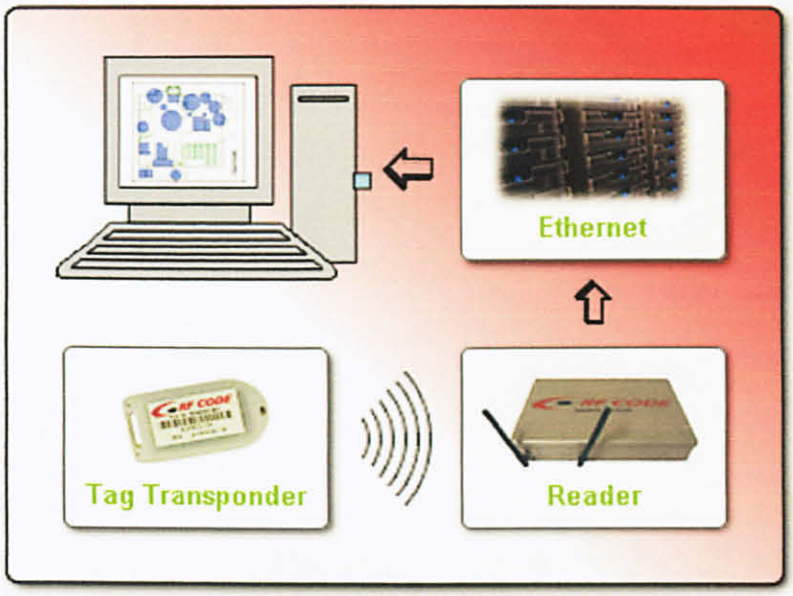

Figure 3: Flow on how active RFID works

Some active tags contain replaceable batteries for years of use; others are sealed units. The major advantages of an active RFID tag are:

- It can be read at distances of one hundred feet or more, greatly improving the utility of the device
- It may have other sensors that can use electricity for power.

The problems and disadvantages of an active RFID tag are:

- The tag cannot function without battery power, which limits the lifetime of the tag.
- The tag is typically more expensive, often costing $20 or more each
- The tag is physically larger, which may limit applications.

- The long-term maintenance costs for an active RFID tag can be greater than those of a passive tag if the batteries are replaced.
- Battery outages in an active tag can result in expensive misreads.

Active RFID tags may have all or some of the following features:

- longest communication range of any tag
- the capability to perform independent monitoring and control
- the capability of initiating communications
- the capability of performing diagnostics
- the highest data bandwidth
- active RFID tags may even be equipped with autonomous networking; the tags autonomously determine the best communication path.

Figure 4: Active RFID tag

2.5 Passive RFID Tag

A passive tag is an RFID tag that does not contain a battery. The power is supplied by the reader. When radio waves from the reader are encountered by a passive RFID tag, the coiled antenna within the tag forms a magnetic field. The tag draws power from it, energizing the circuits in the tag. The tag then sends the information encoded in the tag's memory [8].

The major disadvantages of a passive RFID tag are:

- The tag can be read only at very short distances, typically a few feet at most. This greatly limits the device for certain applications.
- It may not be possible to include sensors that can use electricity for power.
- The tag remains readable for a very long time, even after the product to which the tag is attached has been sold and is no longer being tracked.

The advantages of a passive tag are:

- The tag functions without a battery; these tags have a useful life of twenty years or more.
- The tag is typically much less expensive to manufacture
- The tag is much smaller (some tags are the size of a grain of rice). These tags have almost unlimited applications in consumer goods and other areas.

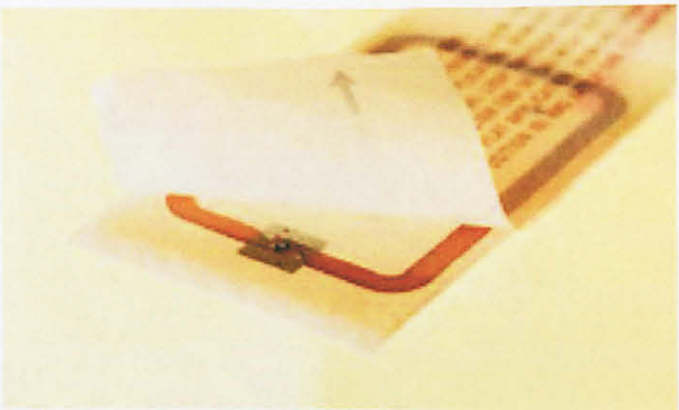

Figure 5: Passive RFID tag

2.6 Comparison between Active and Passive Tag

Criteria\Type of RFID tag	Active	Passive
Cost	Expensive	Cheaper
Communication range	100m or more	3m or less
Availability of tag power	Continuous	Only within field or reader
Tag power source	Internal to tag	Energy transferred from the reader
Available signal strength from tag to reader	High	Low
Data storage	Large read/write data storage (128kB) with sophisticated data search and access capabilities available	Small read/write storage (128 bytes)
Tag battery	Yes	No
Application characteristics	• Dynamic business process • Unconstrained asset movement • Security/sensing	• Rigid business process • Constrained asset movement • Very simple security

Table 1: Differences between active and passive tag [8]

Passive RFID operation requires very strong signals from the reader and the signal strength returned from the tag is constrained to very low levels by the limited energy

[1]. Active RFID allows very low level signal to be received by the tag and the tag can generate high level signal back to the reader [8].

2.7 RFID reader

An RFID reader is a device that is used to interrogate an RFID tag. The reader has an antenna that emits radio waves; the tag responds by sending back its data [6].

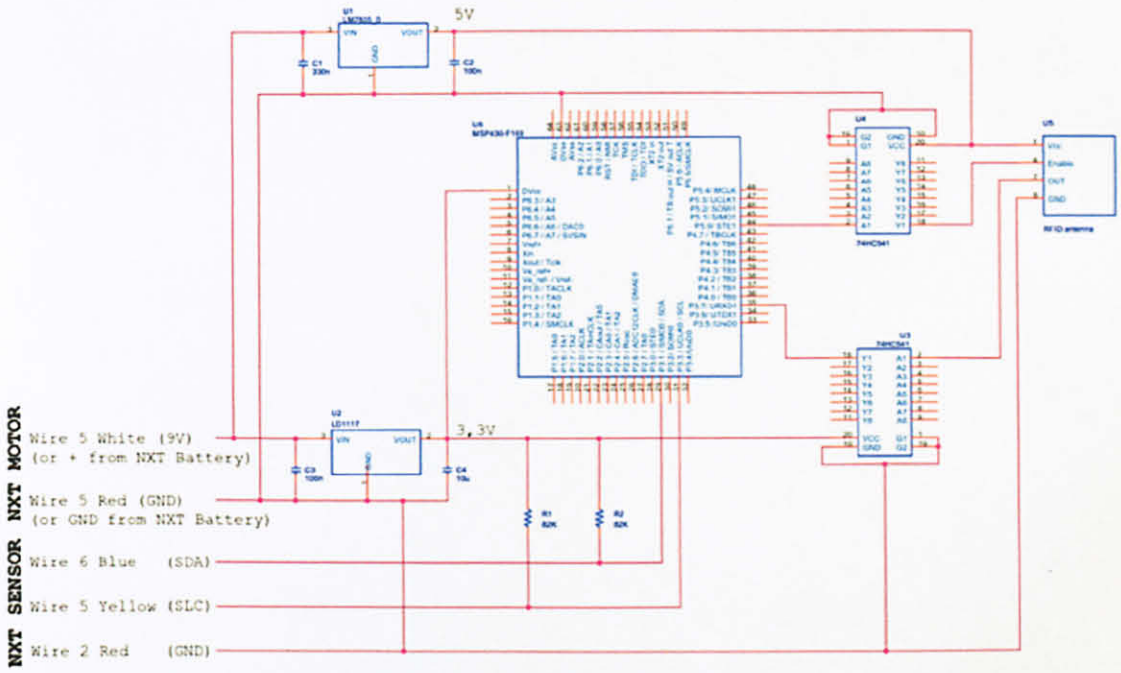

Figure 6: Circuit diagram for RFID reader [9]

There are two types of RFID readers; portable/handheld RFID reader and fixed RFID reader. The handheld RFID reader is more convenient for close-range and manual checking of RFID data while the fixed RFID reader is more ideal for widest range and auto checking the RFID data [2].

2.8 Operating frequencies

Radio-frequency (RF) signals are typically sinusoidal or nearly so - that is, the voltage or field is a smooth, periodic function of time. The number of times the signal repeats itself per second, the frequency, varies widely in differing RFID systems. Frequency is measured in Hertz (Hz): one Hertz is one cycle per second. KHz= 1000's of Hz; MHz = millions of Hz [15].

The figure below shows some of the common and less-common frequency bands in which RFID systems operate. Also shown is the corresponding wavelength - the distance between points at which the field has a fixed value when the signal moves at the velocity of light [15].

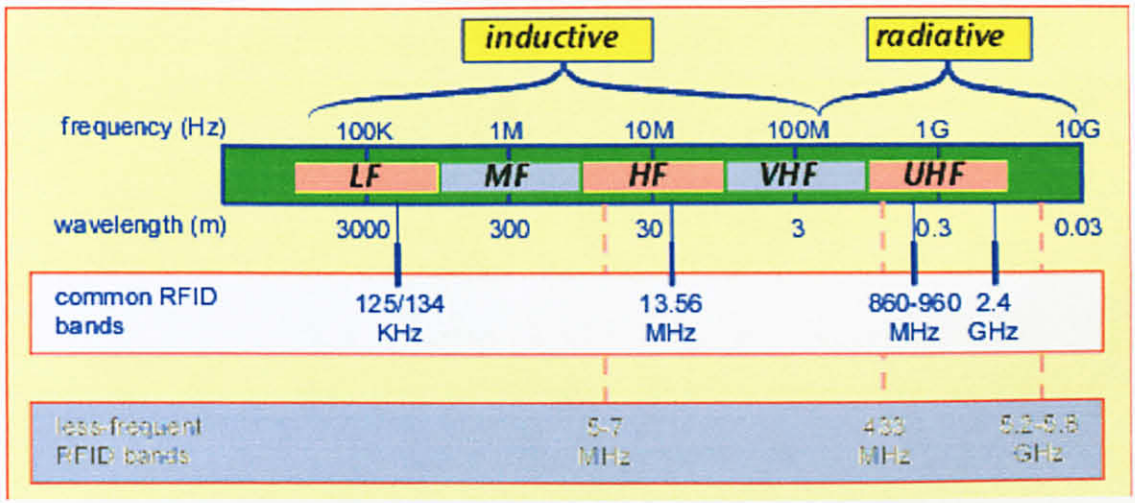

Figure 7: Operating frequencies which RFID systems operate

Several issues are involved in choosing a frequency of operation. The most fundamental, as indicated in the diagram, is whether inductive or radiative coupling will be employed. The distinction is closely related to the side of the antennas to be used relative to the wavelength. When the antennas are very small compared to the wavelength, the effects of the currents flowing in the antenna cancel when viewed from a great distance, so there is no radiation. Only objects so close to the antenna that one part of the antenna appears significantly closer than another part can feel the presence of the current. Thus, these systems, which are known as inductively-

coupled systems, are limited to short ranges comparable to the size of the antenna. In practice, inductive RFID systems usually use antenna sizes from a few cm to a meter or so, and frequencies of 125/134 KHz (LF) or 13.56 MHz (HF). Thus the wavelength (respectively about 2000 or 20 meters) is much longer than the antenna [15].

Radiative systems use antennas comparable in size to the wavelength. The very common 900 MHz range has wavelengths around 33 cm. Reader antennas vary in size from around 10 to >30 cm, and tags are typically 10-18 cm long. These systems use radiative coupling, and are not limited by reader antenna size but by signal propagation issues [15].

A second key issue in selection of frequency bands is the allocation of frequencies by regulatory authorities. In essentially every country in the world, the government either directly regulates the use of the radio spectrum, or delegates that authority to related organizations. In the United States, the ***Federal Communications Commission*** (FCC) regulates the frequencies that radios are allowed to radiate, the power levels they can use, and other more technical aspects of their operation. For much of the history of radio in the US, every radio transmitter needed a license from the FCC to operate the radio. However, in the mid-1980's, the FCC began to allocate certain frequency bands in which ***unlicensed*** operation would be allowed, subject to certain restrictions on the equipment and usage. Many other nations in the world have followed suit. This released a remarkable burst of innovation, including unlicensed cordless phones, wireless local area networks, and other devices. RFID systems are typically operated in unlicensed bands [15].

In the US, unlicensed operation is available in the ***Industrial, Scientific, and Medical*** (ISM) band at 902-928 MHz, among others. The 900-MHz ISM band is a very common frequency range for UHF RFID readers and tags. It is important to note, however, that bands do not exist in isolation; for example, the figure below shows the various uses to which frequencies near the ISM band are put in the United States [15].

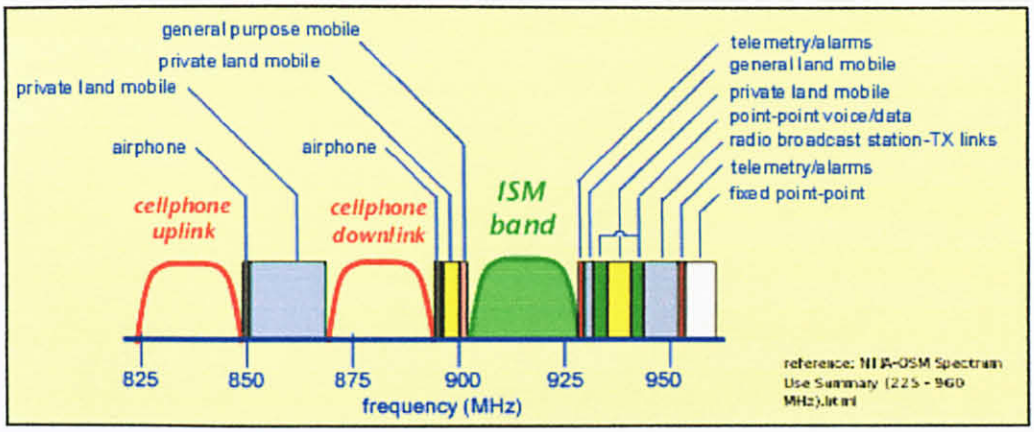

Figure 8: Various uses frequencies near ISM band are put in US

The practical consequence of this proximity is the possibility of ***interference***: for example, a nearby cell phone transmitting tower may interfere with the operation of RFID readers, due to the finite ability of the reader receiver to reject the powerful cell signal. (Cellular base stations may sometimes use transmit powers of 10's to hundreds of watts.) Other users of the ISM band may also interfere with RFID readers, or encounter interference due to them: examples are cordless phones and older wireless local area networks [15].

Finally, changes in operating frequency affect the propagation characteristics of the resulting radiated fields. Lower frequencies diffract more readily around obstacles, but couple less well to small antennas. Radiated fields are absorbed by many common materials in buildings and the environment, particularly those containing water. The degree of absorption due to water increases gradually with increasing frequency. Tags immersed in water-containing materials (i.e. injected into or swallowed by animals or people) must use very low frequencies to minimize absorption: this is a typical 125 KHz application. For locating large objects or people outdoors, a relatively low frequency (e.g. 433 MHz) may be desirable to avoid obstacle blockage; when a clear line of sight from the antenna to the tag can be assured, a higher frequency may be useful to reduce the size of the antennas [15].

2.9 Operating principles for RFID

RFID has many different principles of operating systems. There are two basic important principles for the RFID to operate; inductive coupling and backscatter coupling [4].

2.9.1. Inductive Coupling

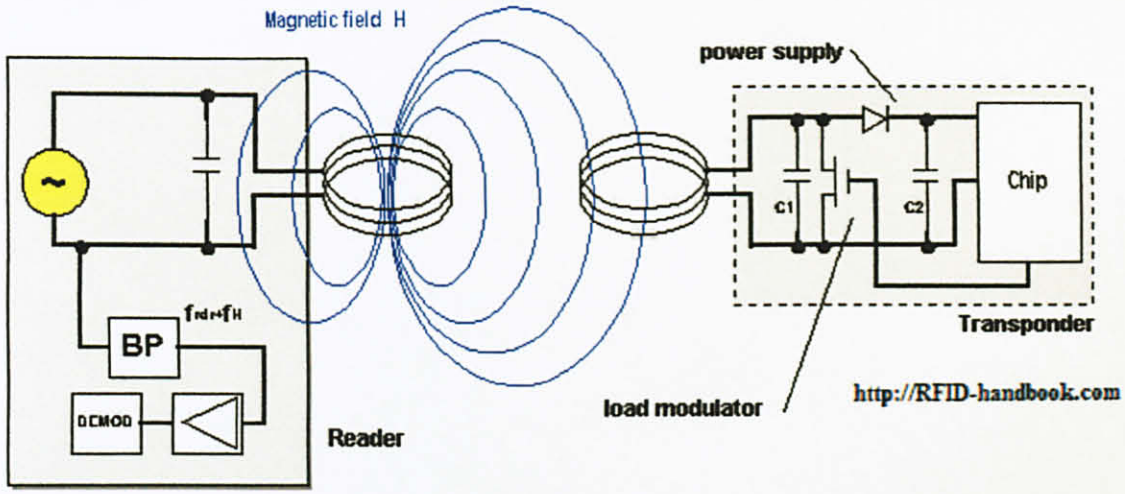

Figure 9: Inductive coupling system

Inductive coupling transponder is almost operated passively. As described above, inductively coupled systems are based upon a transformer-type coupling between the primary coil in the reader and the secondary coil in the transponder. This is true when the distance between the coils does not exceed 0.16 l, so that the transponder is located in the near field of the transmitter antenna [4].

2.9.2. Backscatter Coupling

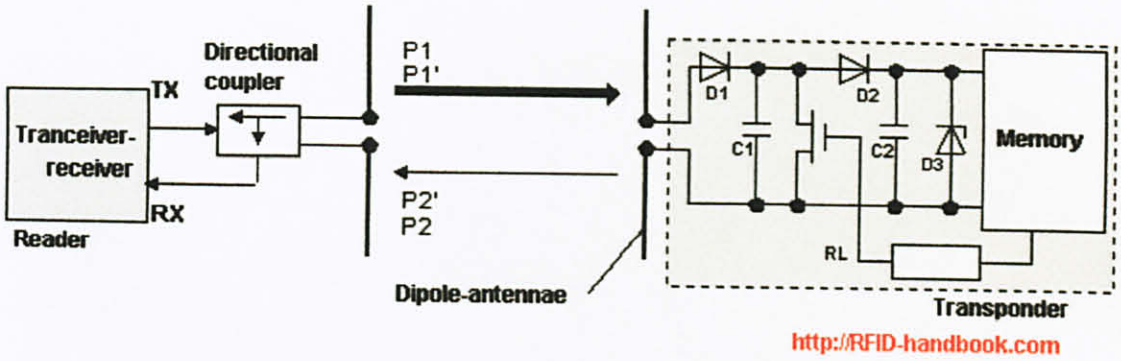

Figure 10: Backscatter coupling system

Power P1 is emitted from the reader's antenna, a small proportion of which reaches the transponder's antenna. The power P1' is supplied to the antenna connections as HF voltage and after rectification by the diodes D1 and D2 this can be used as turn on voltage for the deactivation or activation of the power saving "power-down" mode. The diodes used here are low barrier Schottky diodes, which have a particularly low threshold voltage. The voltage obtained may also be sufficient to serve as a power supply for short ranges. A proportion of the incoming power P1' is reflected by the antenna and returned as power P2. The reflection characteristics of the antenna can be influenced by altering the load connected to the antenna. In order to transmit data from the transponder to the reader, a load resistor RL connected in parallel with the antenna is switched on and off in time with the data stream to be transmitted. The amplitude of the power P2 reflected from the transponder can be modulated. The power P2 reflected from the transponder is radiated into free space. A small proportion of this is picked up by the reader's antenna. The reflected signal therefore travels into the antenna connection of the reader in the "backwards direction" and can be decoupled using a directional coupler and transferred to the receiver input of a reader. The "forward" signal of the transmitter, which is stronger by powers of ten, is to a large degree suppressed by the directional coupler [4].

2.10 RFID applications

Universiti Utara Malaysia is using the newest way for taking the class attendance by using RFID and this method has lighten the burden for the lecturer to key in the record of the student's attendance. It is also has decreased the percentage of student from escaping their class.

Features	Present Application	Project
Function	As a attendance collector	Track the student either they are in the class or not
Security	N/A	Thumbprint Identification Technology

Table 2: Difference between present application and project

CHAPTER 3

METHODOLOGY

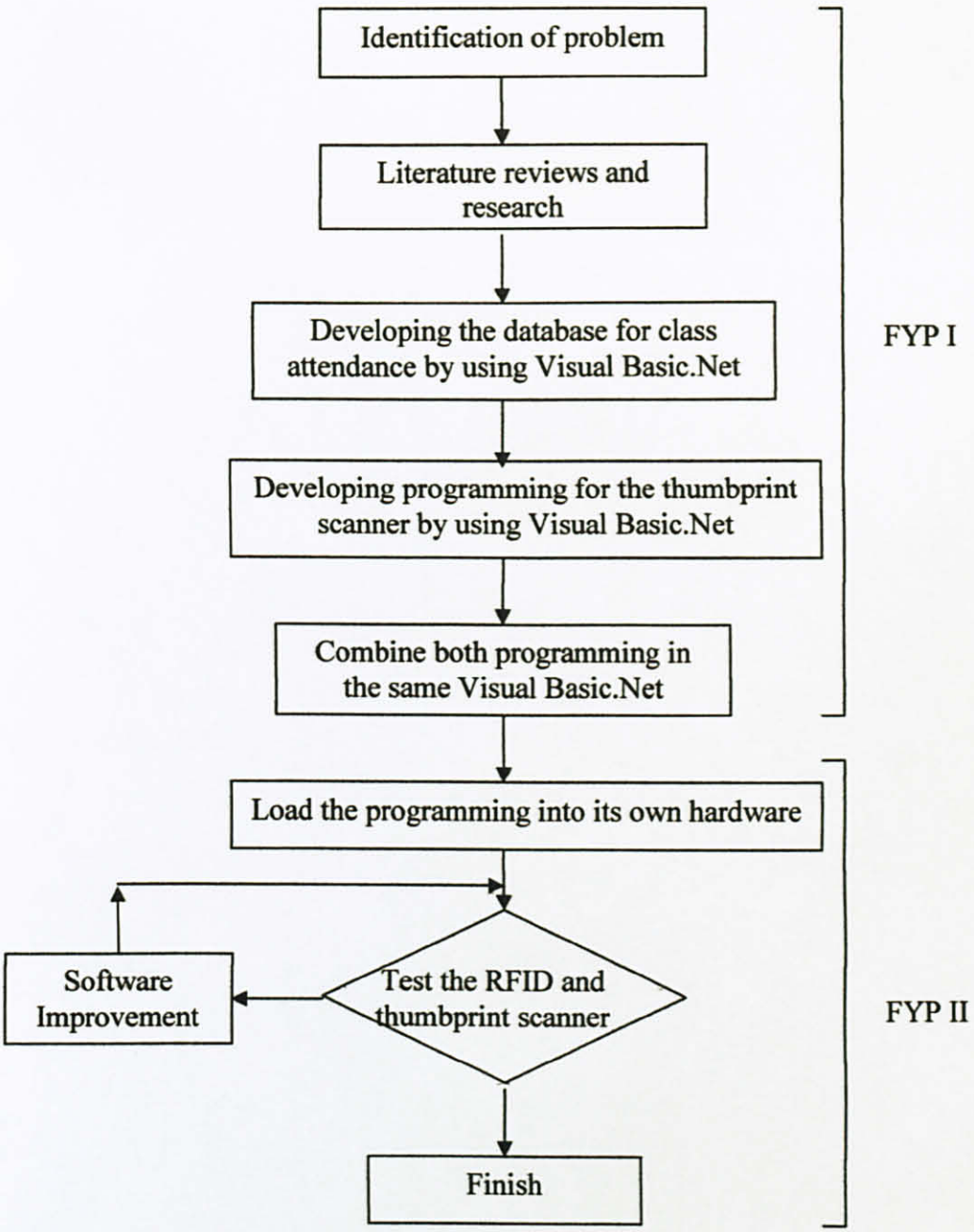

Figure11: Flow chart of the project

3.1 Procedure Identification

3.1.1 FYP I

This stage is the beginning of project's development. Problem identification, research and discussion regarding the project have been done during this period. Complete research and better understanding regarding the project is a must to get a better result.

After all the research has been done, the development of the programming and hardware need to be done. The completed programming needed to be tested to make sure the system is running. The interfaces and database have to be completed during this stage. The specifications about the reader and active tag have to be considered.

3.1.2 FYP II

Finally, the last stage that needed to be focused more at the fabrication of the hardware. At this stage, the software needed to load into the hardware and test needed to be done.

3.2 Tools and Hardware

3.2.1. Active RFID tags

This device will be attached with student's matrix card.

3.2.2. RFID reader

The reader will be put in the lecture hall or tutorial room. It will be located anywhere either at the back or in front of the classroom.

3.3 Software

3.3.1. Visual Basic.Net

The coding and database that have been generated by Microsoft Access will transfer to this software. The RFID and thumbprint scanner will be connected to this software.

3.2.2 Programming Station

This software is being used to online the reader. It is the connector between the reader and the network. The tag's calling process is also in this system

3.4 Basic concept of the project

The main idea of this project is to use the integrating of Radio-Frequency Identification (RFID) and thumbprint scanner as a class attendance. The thumbprint scanner will act as an attendance collector and the RFID will be the checker either the student is in the class or not. Its basic concept is shown in the figure below.

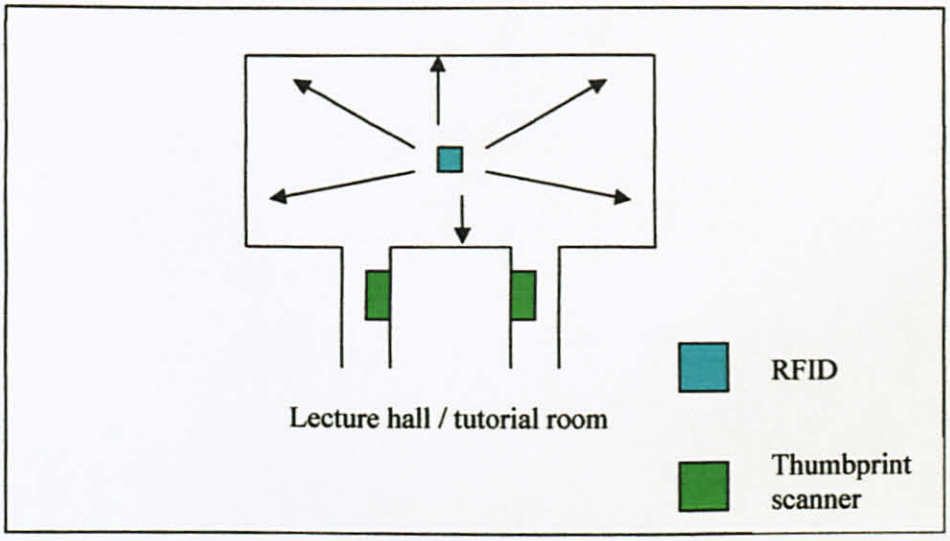

Figure 12: The basic concept of the project

Student will come to the class and use their thumbprint to replace the signature. Then I added RFID as a tracker to make sure the student is in the class. The RFID has its own range to read and detect their tags. This device can be a good device to read the attendances of the students in the class. The thumbprint scanner will be located at the door while the RFID reader will be located at the back of the lecture hall or tutorial room. The active RFID tag will be put at the back of student's matrix card. The tags will send a signal to the reader and the reader will read the signal and send the data to the database.

3.5 Overview of the Project

Student will scan their thumbprint before entering the class

Student enter the class

Active tag that have been attached behind student's matric card

Tag send signal to reader

Reader receive signal and send signal to tag

Reader sends data to computer

Computer host stores the database

Figure 13: Overview of the project

CHAPTER 4
RESULTS AND DISCUSSION

This project consists of two main parts; software and hardware. As for this project, the programming has been created using Visual Basic.Net 6 while the hardware has been supplied by activeWACE INC. The specifications of the hardware have been attached in Appendix L and Appendix M.

4.1 Software

As for this project, the simulator of thumbprint scanner has been used to demonstrate the functioning of the system.

4.1.1 Database

The database for the system has been created by using the Microsoft Access.

Student

ID	Student ID	Thumbprint	RFID	Name	MobilePhone	EmailAddress
1	8667	222	581	Anith Safura bt Azmi		
2	8696	223	582	Ili Nadiah bt Mhd Nasir		
3	8292	224	584	Nurul Atiqah bt Mat Ayu		
4	8298	225	585	Faizatul Khasanah bt Su		
5	7878	226	586	Ahmad Afiq b Mohd Sai		
6	8643	227	587	Munirah bt A.Jabar		
7	7764	228	588	Siti Nurbalqish bt Samin		
8	8003	229	589	Mohd Rezza Bin Rahma		
9	6009	230	590	Farid Aqmal Bin Azmi		
10	15001	231	591	Mohd Iqbal Bin Azmi		
11	10033	232	592	Amy Samiah bt Zainudd		
12	12006	233	593	Azmi b Aziz		
*						

Figure 14: Database for the system

The student's id, student's thumbprint id, student's RFID id have been assigned in the database.

4.1.2 Thumbprint Simulator

The simulator for thumbprint scanner has been created by using the Microsoft Visual Basic 6.

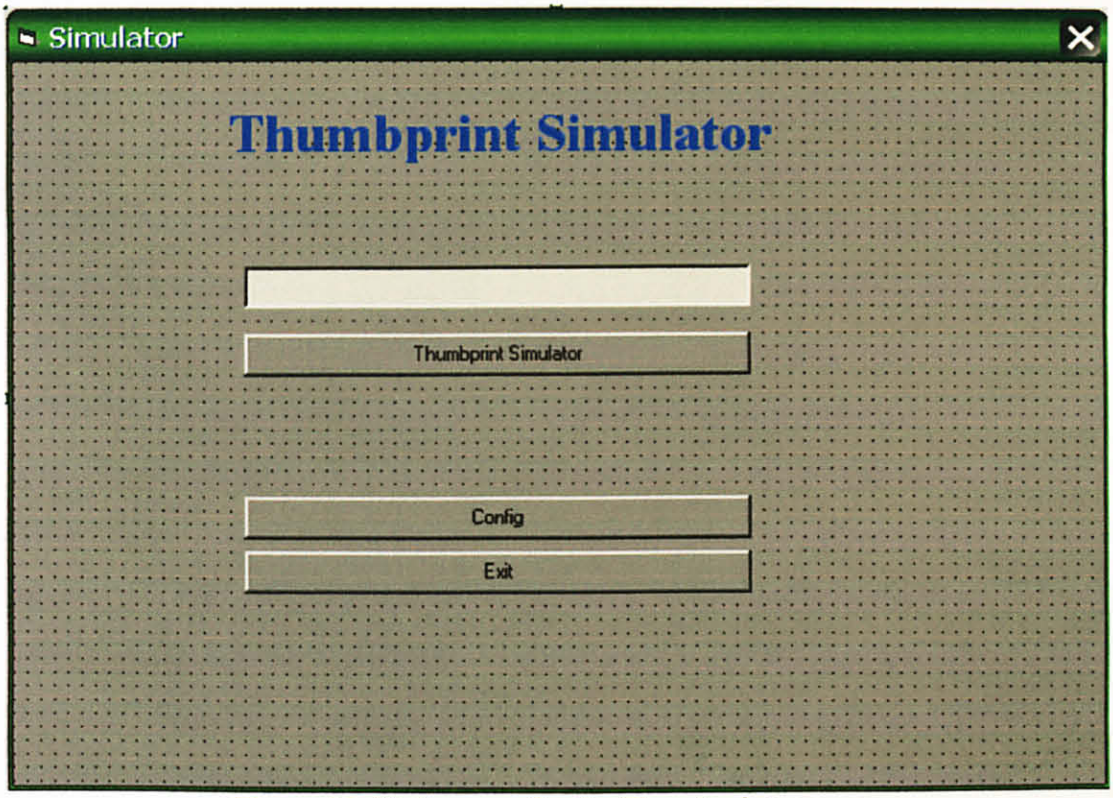

Figure 15: Main menu for the Thumbprint scanner

Figure 16: Thumbprint simulator button

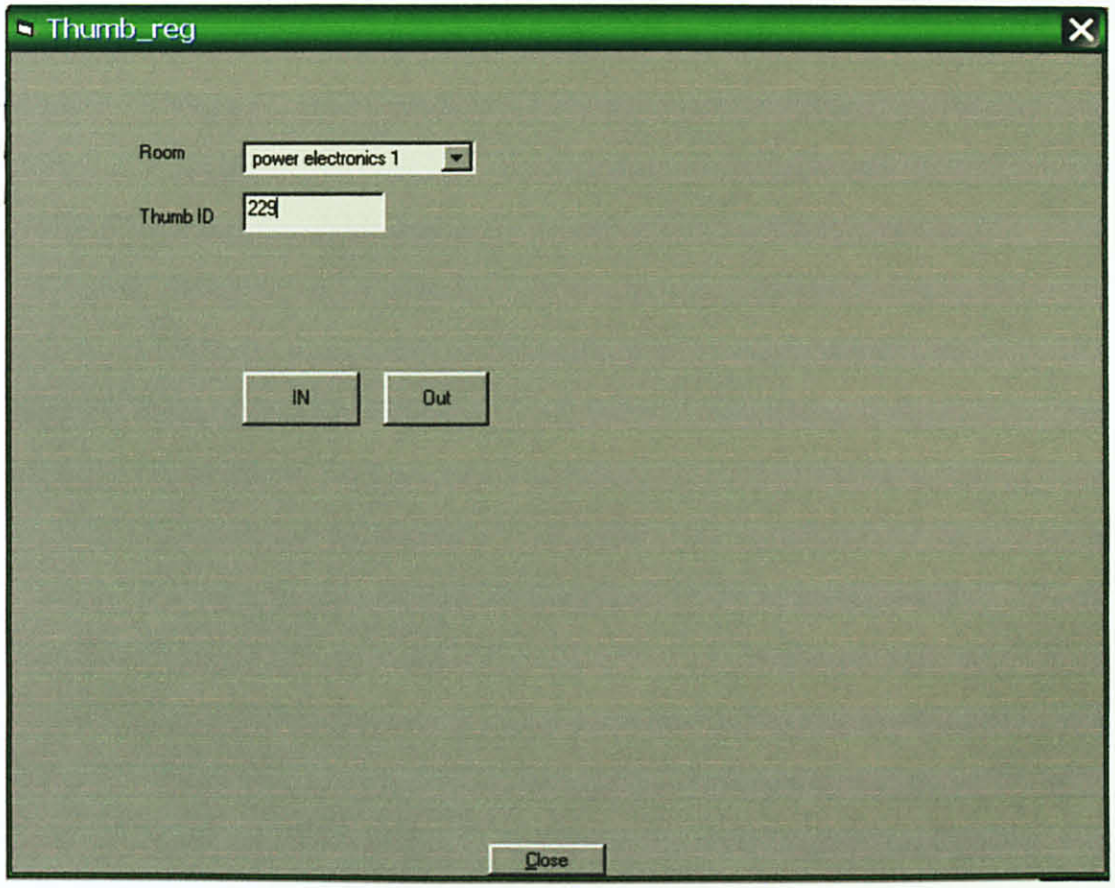

Figure 17: Thumbprint registration

The thumbprint registration will pop-out when the thumbprint simulator button is been clicked. Each student has their own student's thumbprint id. Insert the thumbprint id in the specified room then click on IN button and pop-out window will show that the inserted thumbprint id has been stored into the database. The student's name that has the inserted Thumb ID will be appeared in the database of class attendance.

For example, based from the database, Thumb ID 229 is belonging to Mohd Rezza Bin Rahmat and the room is at the power electronics 1' class. Figure 18 shows the results when we view the attendaces where Mohd Rezza's name has appeared in the attendance list.

Figure 18: Pop-out window for Thumb ID

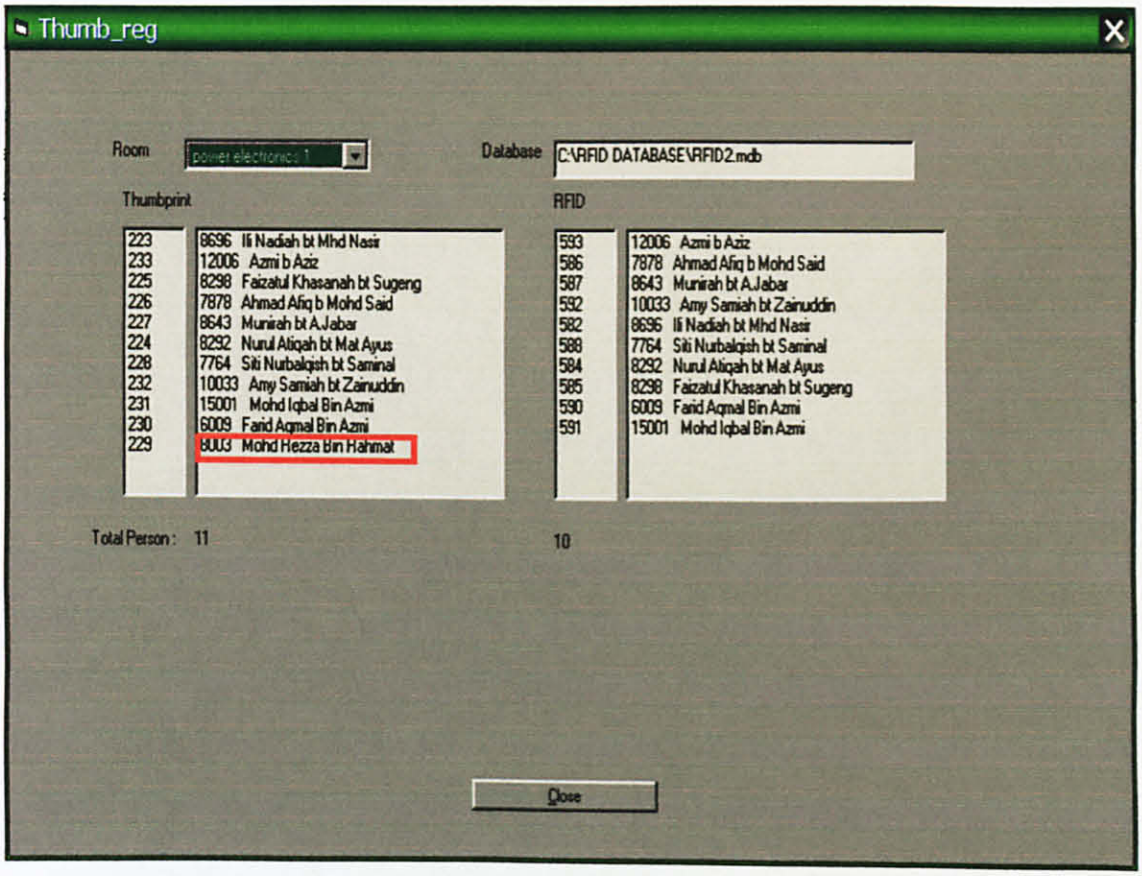

Figure 19: Mohd Rezza's name appeared in the attendance list

4.1.3 RFID Simulator

The RFID simulator has been created by using Microsoft Visual Basic 6.

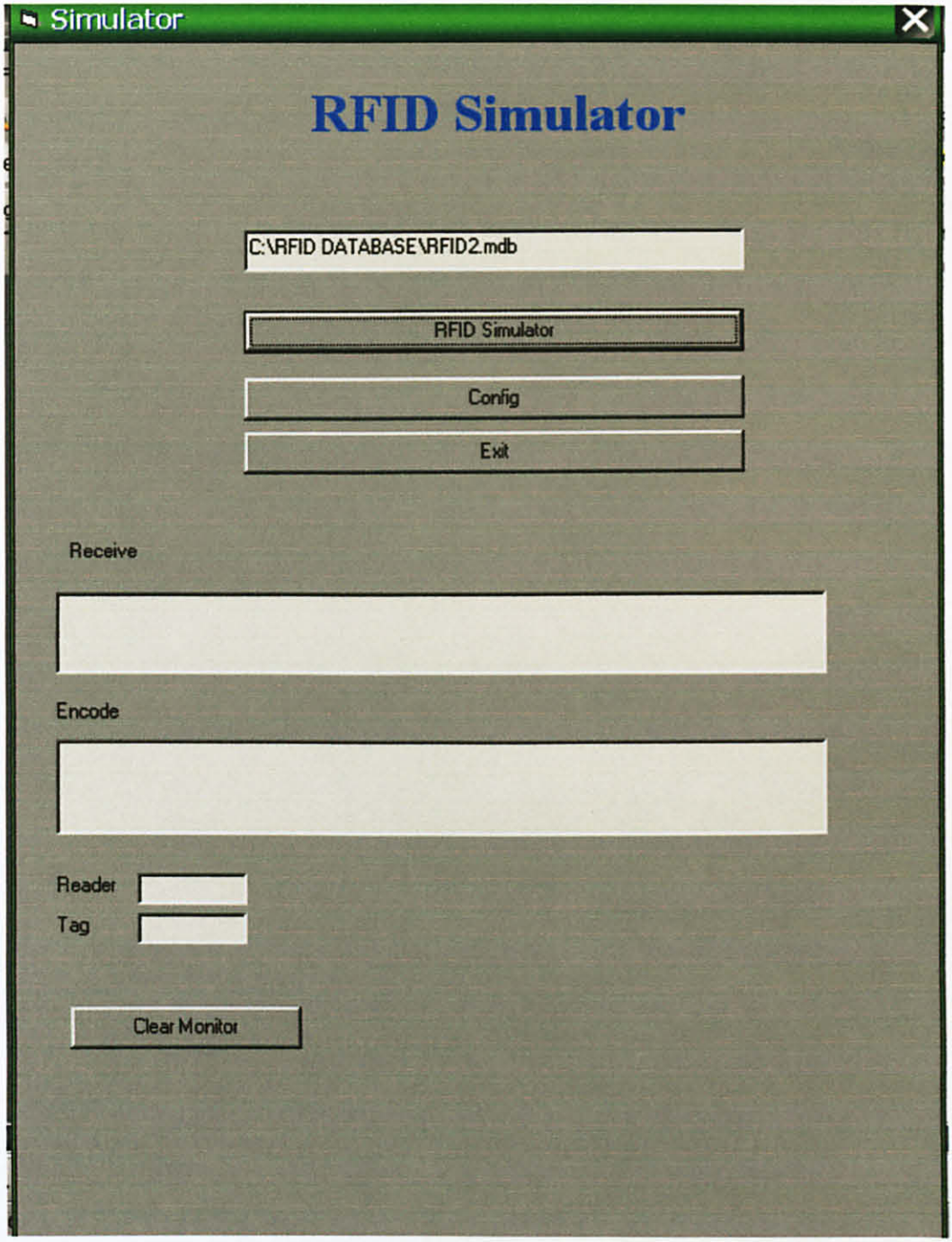

Figure 20: RFID Simulator

Figure 21: RFID simulator button

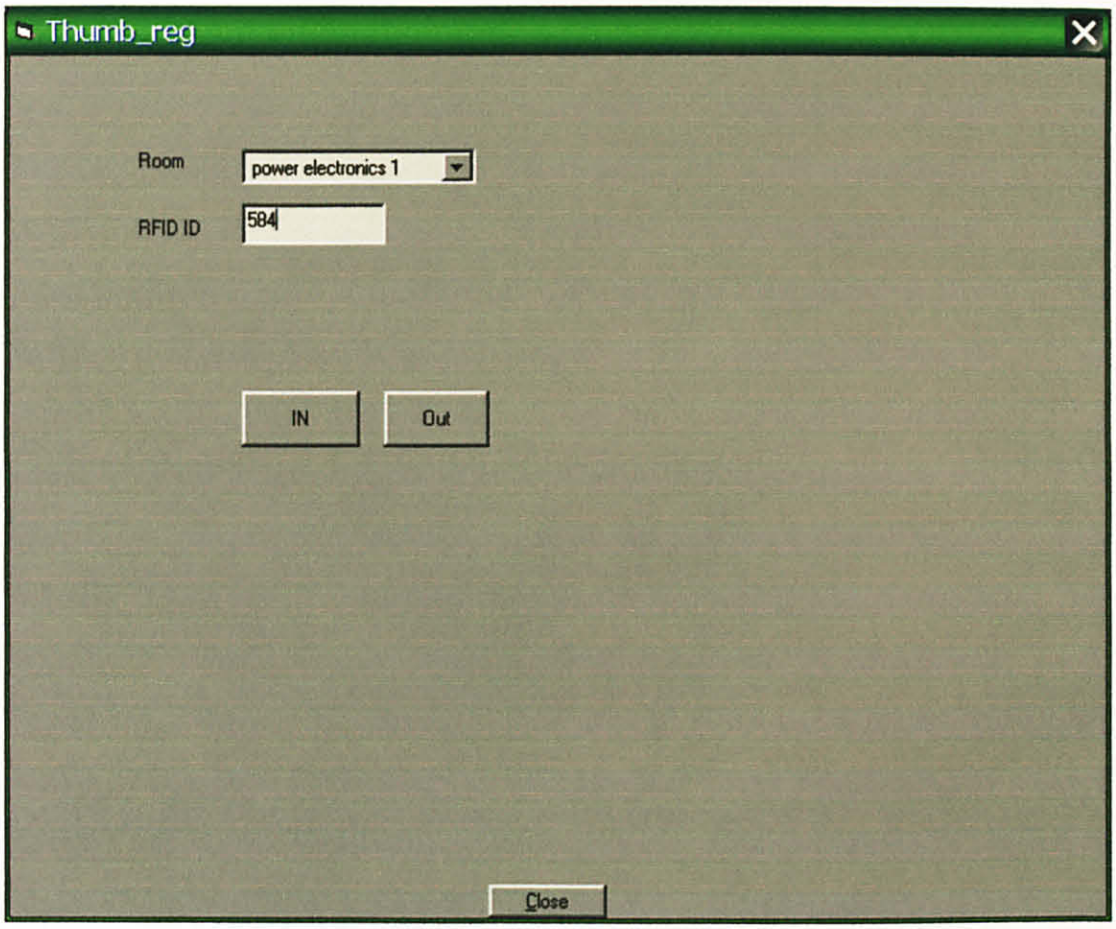

Figure 22: RFID registration

RFID registration window will pop-out when the RFID simulator is been clicked. It is functioning just like the thumbprint scanner simulator.

For example, based from the database, RFID ID 584 is belonging to Nurul Atiqah Bt Mat Ayus. When she is entering the class then the tag will send a signal to reader and reader will detect. Pop-out window will appeared to show that the data has been stored into the database. Then Nurul Atiqah's name will appeared in the attendance list.

Figure 23: Pop-out window for RFID ID

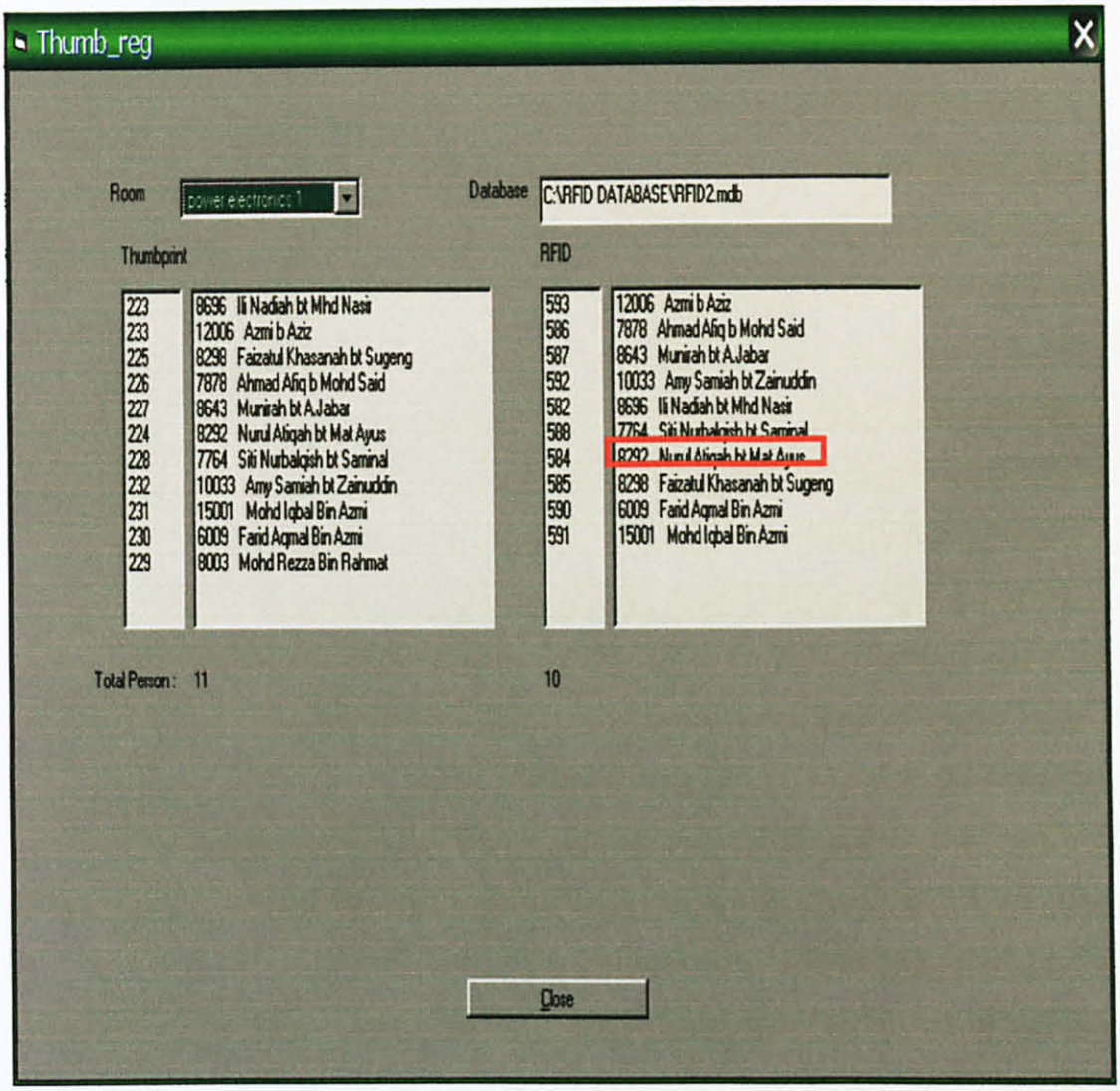

Figure 24: Nurul Atiqah's name appeared in class attendance list

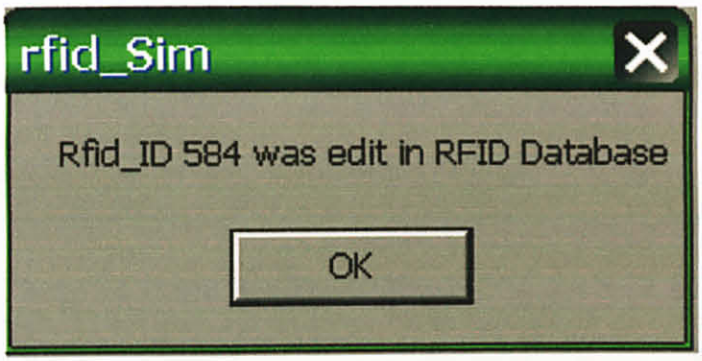

Figure 23: Pop-out window for RFID ID

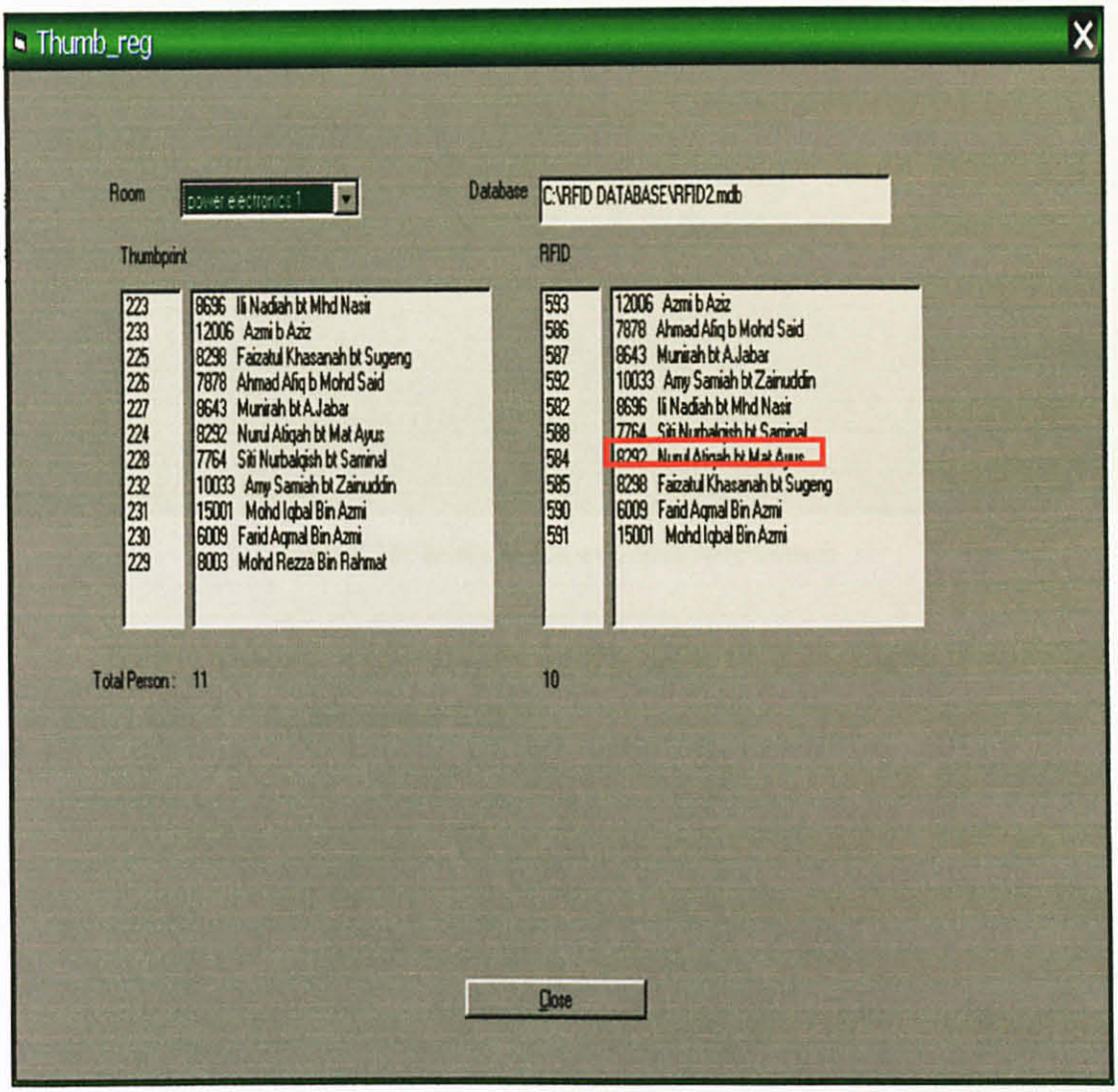

Figure 24: Nurul Atiqah's name appeared in class attendance list

4.1.4 Main Menu of Class Attendance

This is the main interface of the project. This interface has been created by using the Microsoft Visual Basic 6.

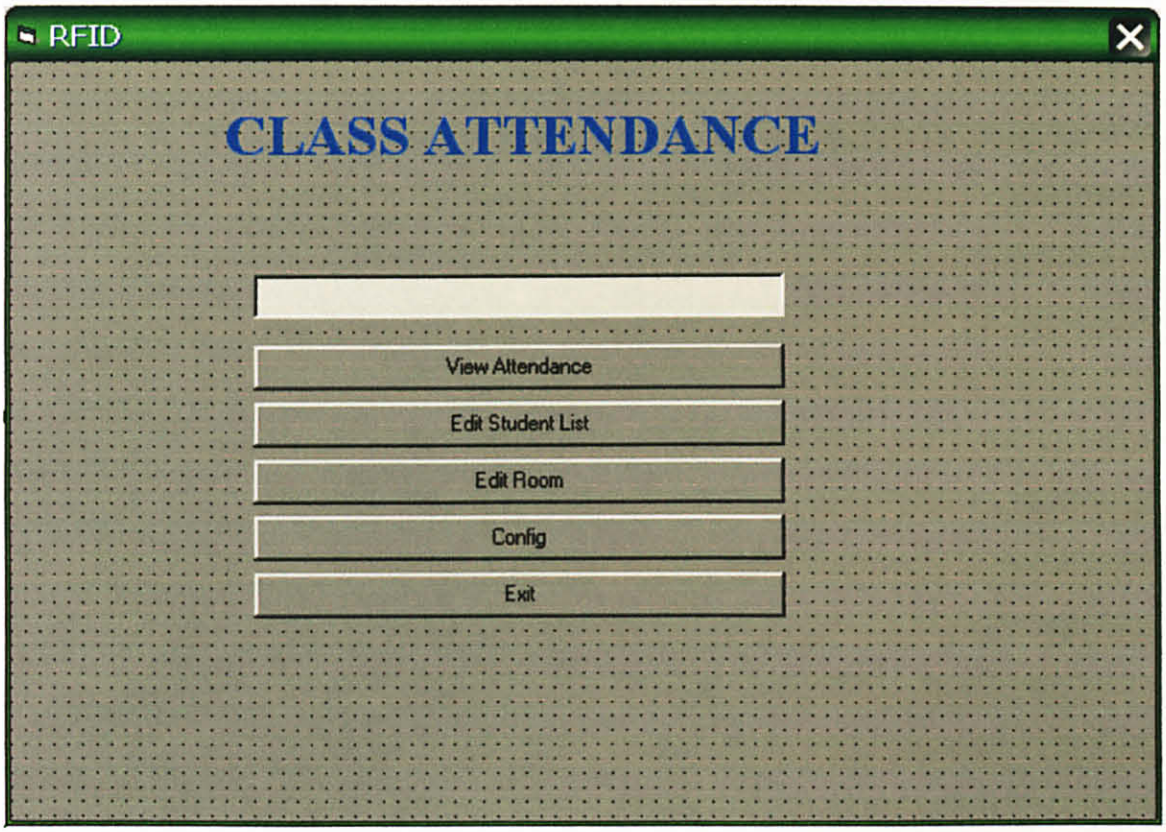

Figure 25: Main Menu of Class Attendance

On this interface, it consists of 5 main buttons; View attendance, Edit student list, Edit room, Configuration and Exit.

As for View attendance, the entire class attendances that have been recorded are being stored here.

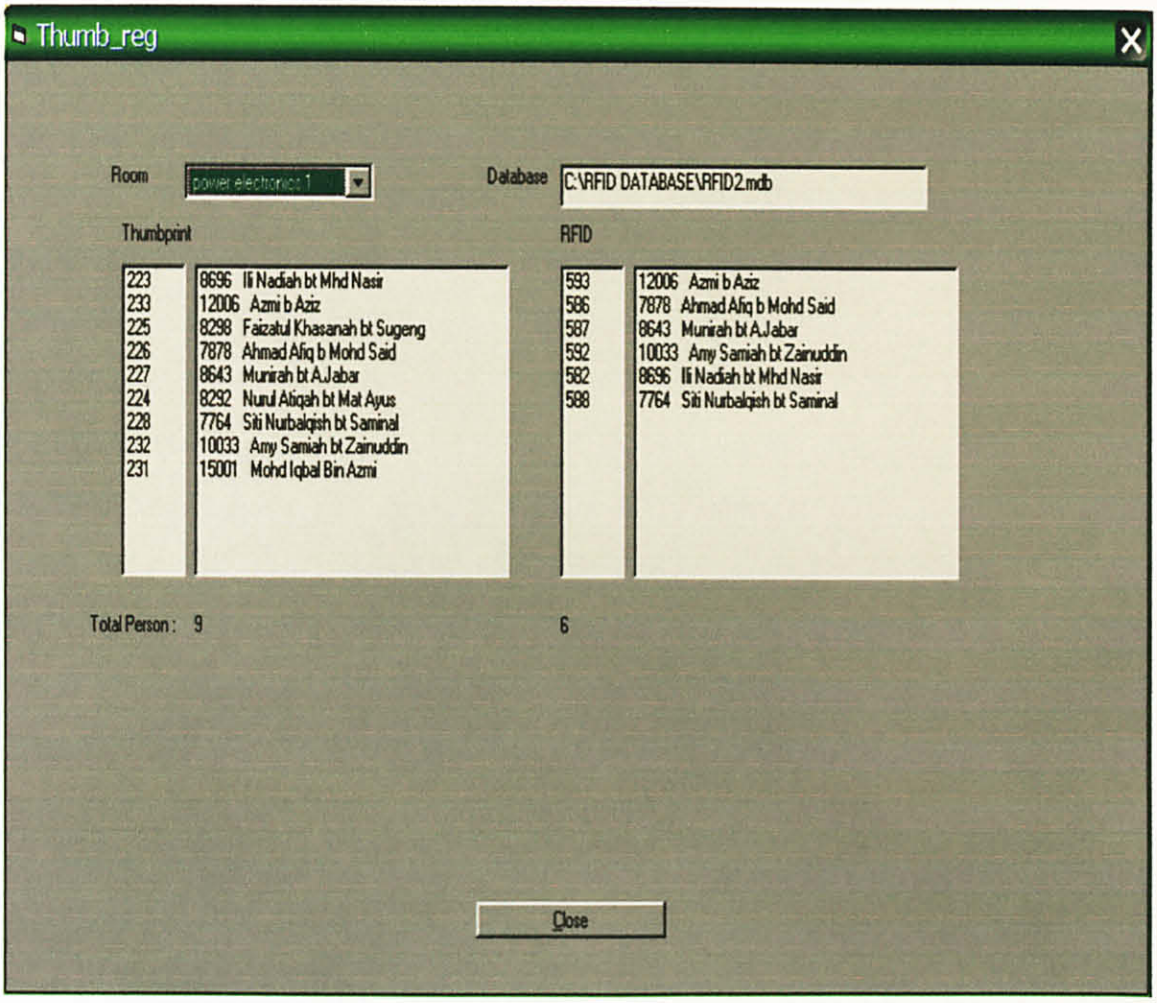

Figure 26: Example of the attendance

From the figure above, there are a difference number of total person for the thumbprint and RFID. It shows that some students just scanned their thumbprint but do not come and listened to the lecture while some students were scanning their thumbprint and coming into the class. For lecturers to observe the attendances, he or she just takes the name that has been displayed in both sides.

As for Edit Student List, the lecturers can edit the information of the students just from here. The same function goes to the Edit Room button. The lecturers can add the venue of the classroom.

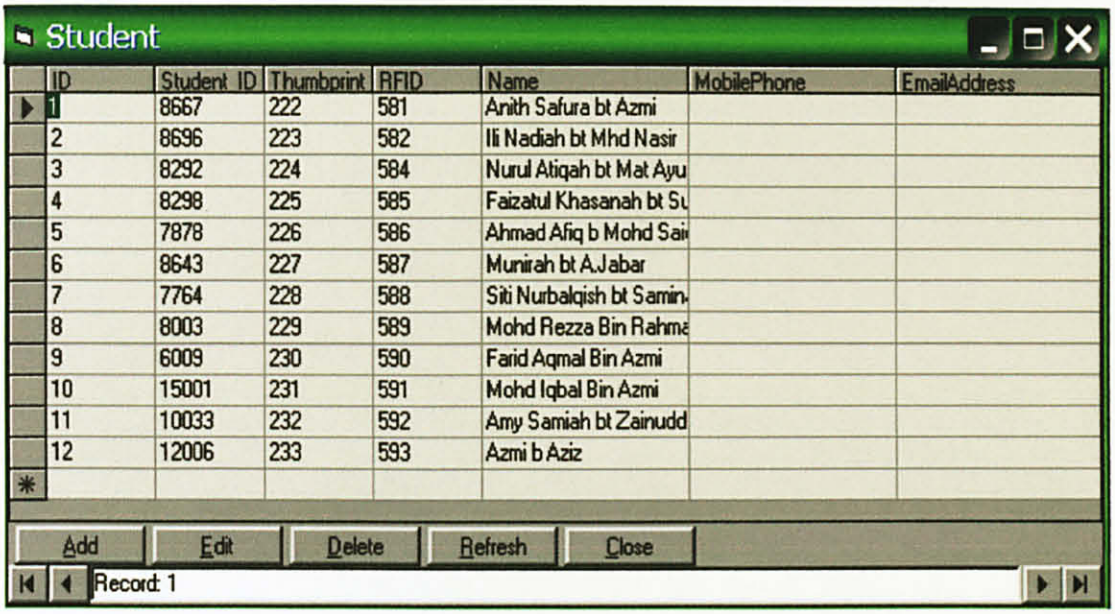

Figure 27: Student list

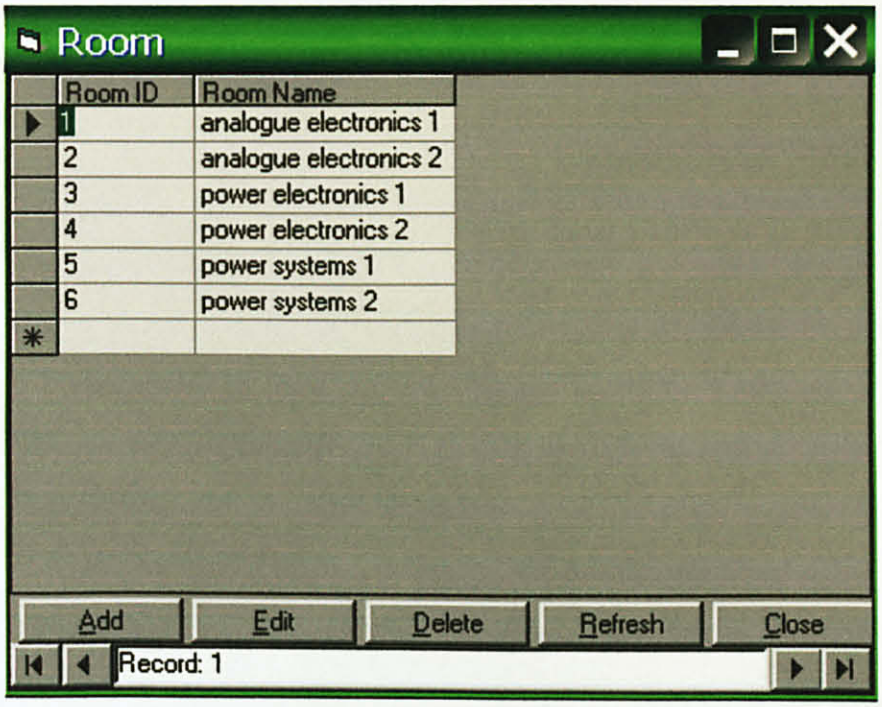

Figure 28: Room list

The function of the Configuration is to configure any database that we want to search. We can see the database from the different room or network that has the same system. We just need to browse and select the desired database.

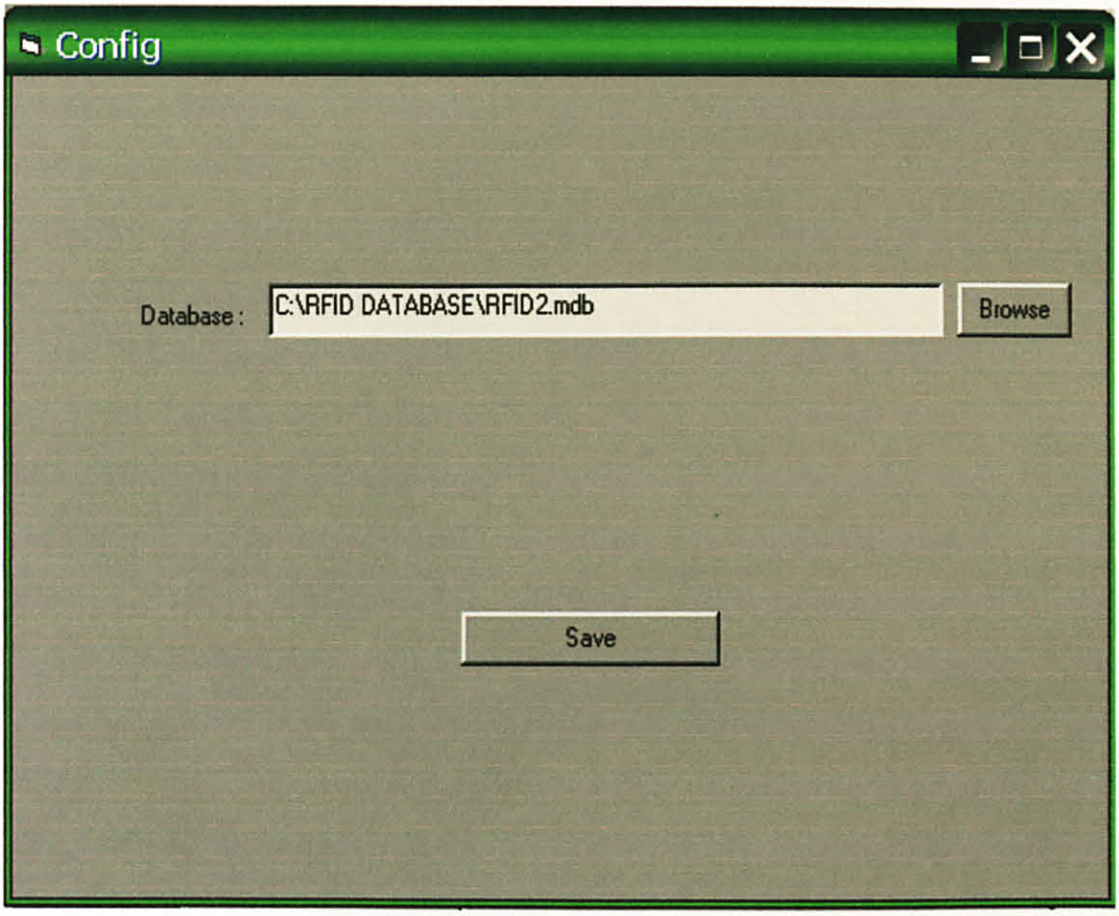

Figure 29: Configuration

When we clicked on the browse button, then figure below will pop-out. Then we can select the file from other computer that has the same software and system.

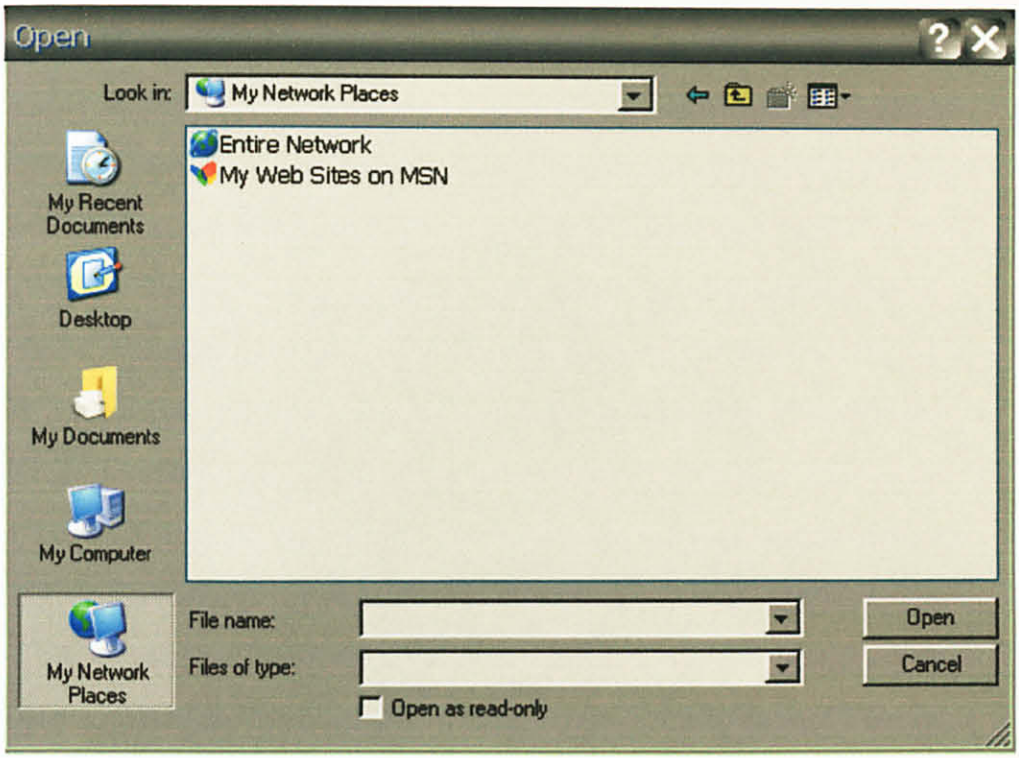

Figure 30: Pop-out window to search for other attendances

4.2 Hardware

Hardware for this project is;

1. Active tag : To attached behind the student's card. Every tag has its own id. It will send a signal to the reader in certain range.
2. RFID reader : Located inside the classroom. It will detect the signal from the tag and send to the database.
3. RS232 cable : Connector between the readers to the database.
4. LAN cable : Connector between the readers to the network.
5. Programming Station : Software needed to online the reader and calls the tags.

Figure below shows the connection between the hardware.

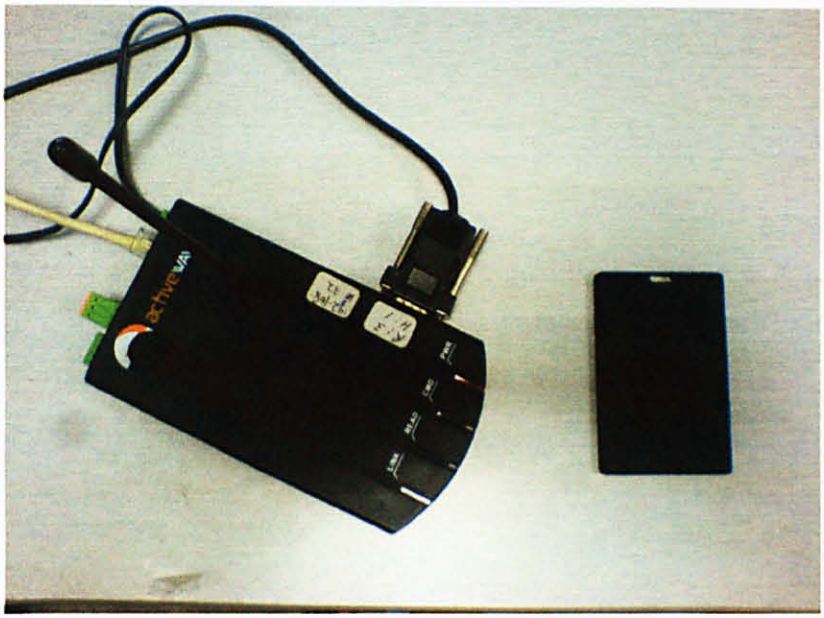

Figure 31: Connection between the hardware.

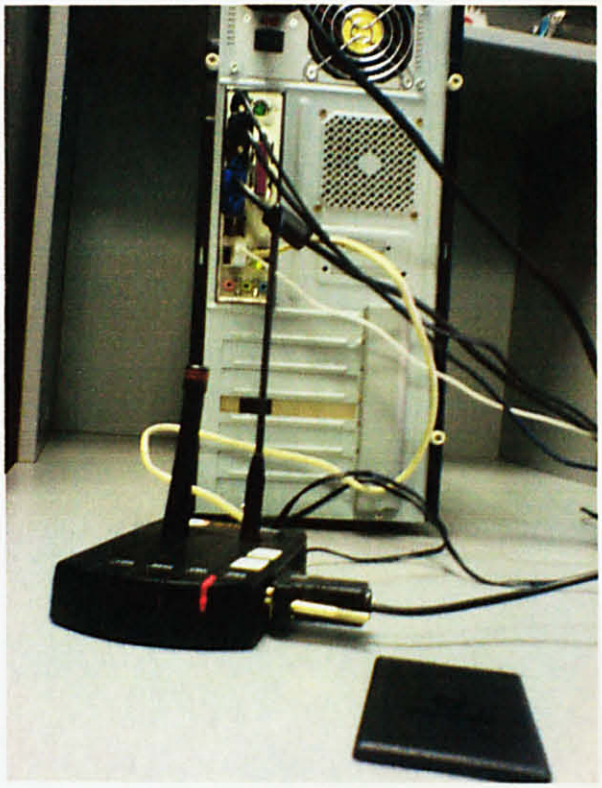

Figure 32: Connection between the reader to the computer

4.3 Combination of hardware and software

The hardware and software needed to be connected together by using RS232 wire. RS-232 (Recommended Standard 232) is a Standard American format for serial data transmission by cable, such as from computer terminal to modem. The Programming Station also has been used to online the reader.

First, the connection between the reader and computer needed to be done by using RS232 and LAN cable. Figure 31 has shows the connection between the reader and computer. On the power supply of the reader and it is ready to be used. Then, the IP address of the computer needed to be changed at Internet Protocol (TCP/IP) Protocol in Local Area Connection to 192.168.2.27.

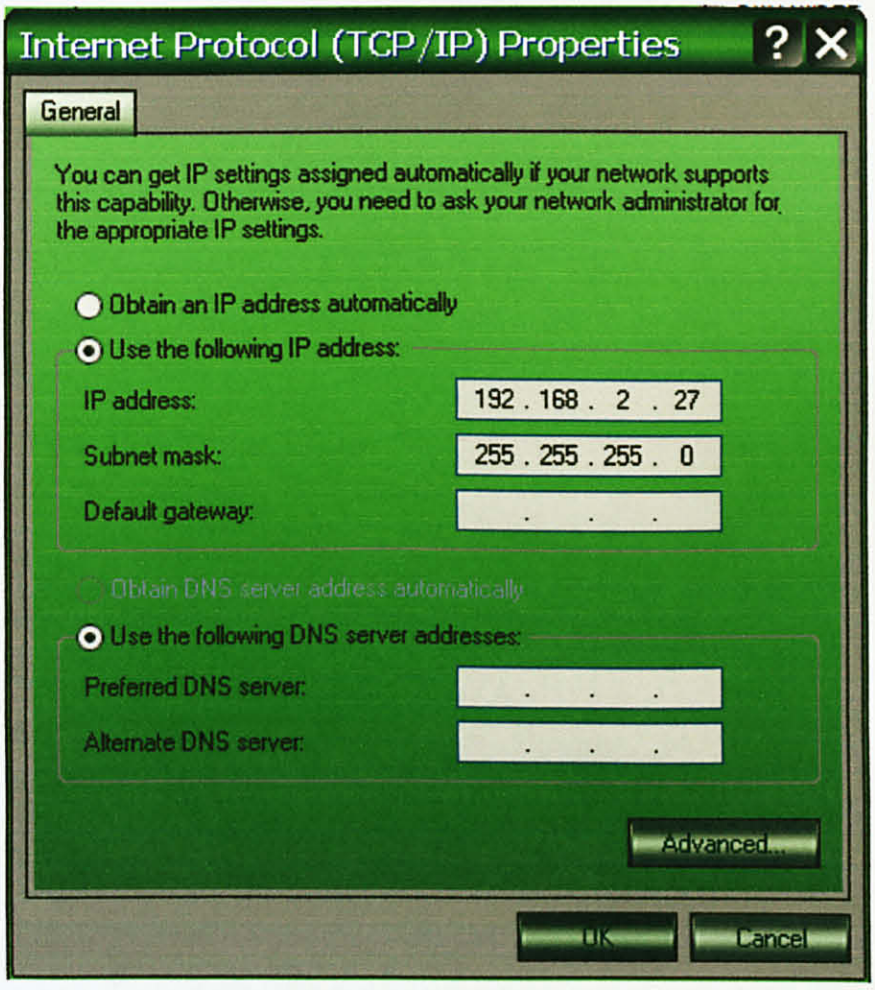

Figure 33: IP address of the computer

The IP address of the reader need to be added at the Programming Station before it is online.

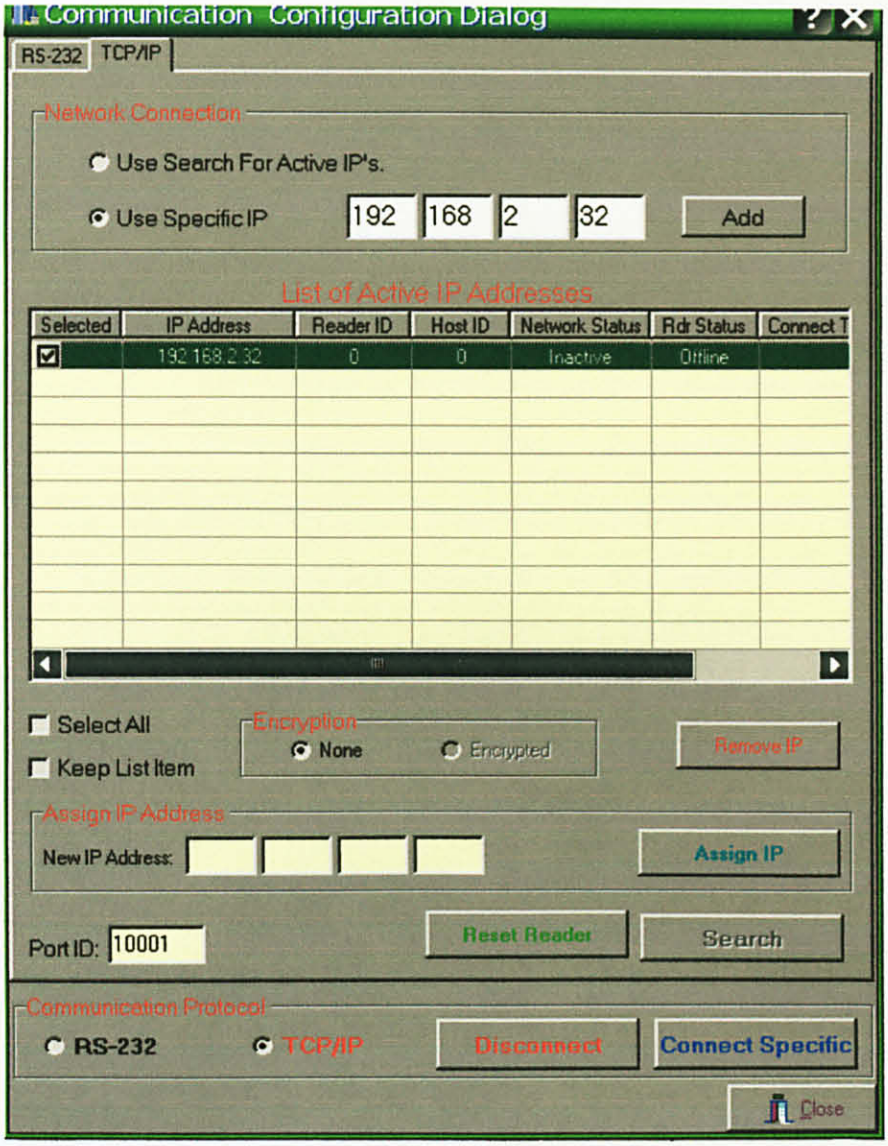

Figure 34: IP address of the reader

After we clicked the Connect Specific button, the Network status will changed from Inactive to Active and Rdr Status (Reader status) will also changed from Offline to Online. This shows that the reader has successfully connected to the computer and can be used during that time. This process needed to be done to make sure that all the

data that have been transmitted to the reader can be sending and stored into the database in the computer.

In order to receive a signal from the tag then the Call button must be clicked and the pop-out window will come out to confirm the operations.

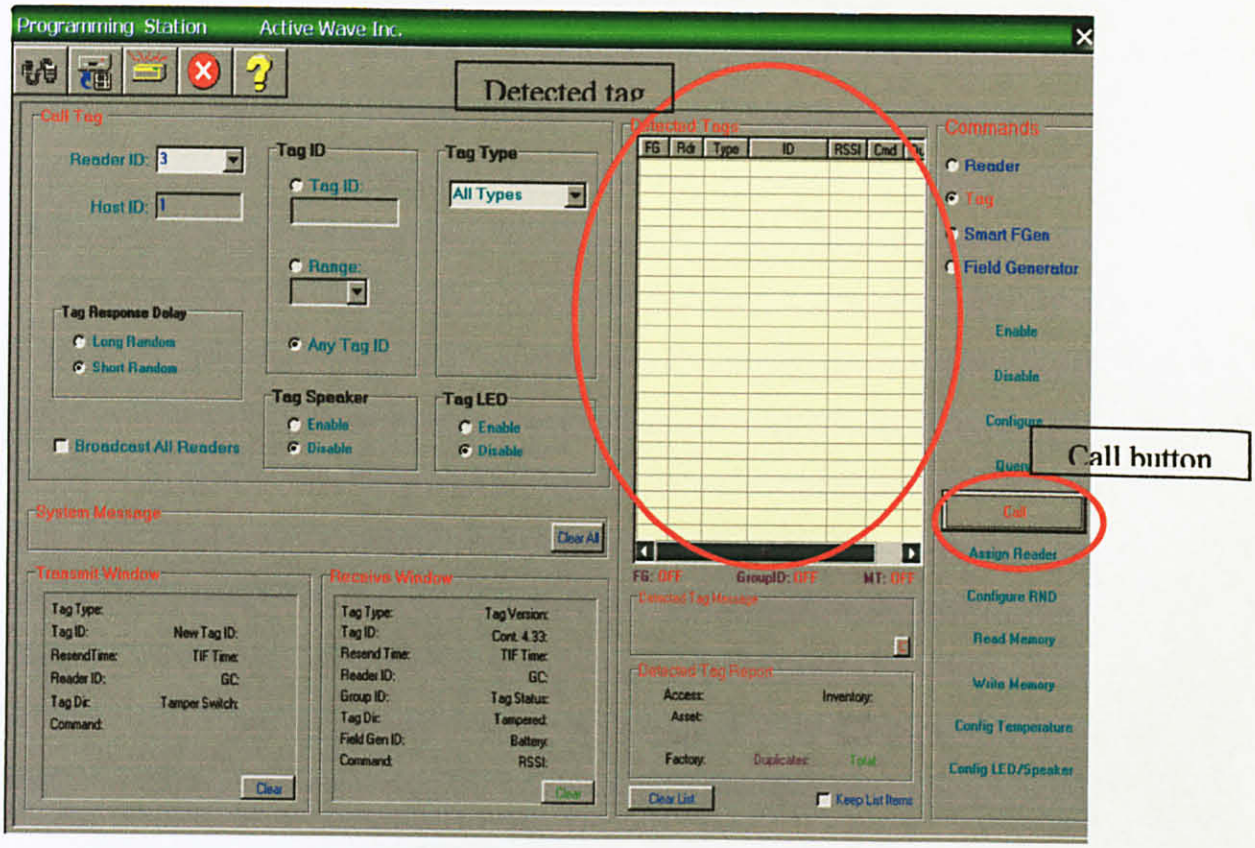

Figure 35: Calling the tags

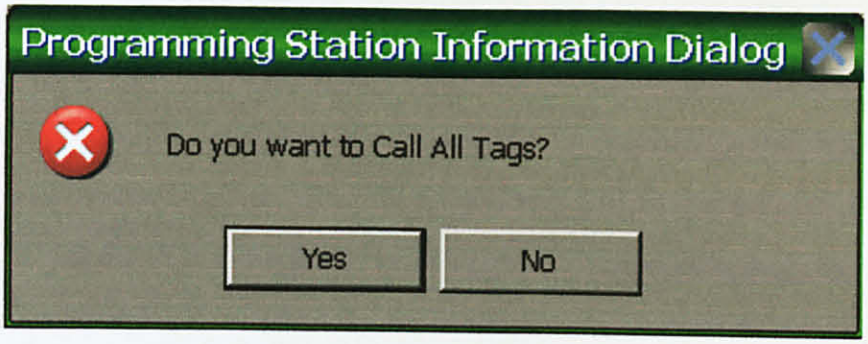

Figure 36: Confirmation pop-out window

When the reader detects the tag in a certain range then the list of the tags will be appeared at the Detected tag place.

As for this project, the tag that has been used is 584. When the reader has detected the reader then it will send the data to the database and the pop-out window will come out and shows that RFID ID 584 has been stored into the class attendance list.

Figure 37: Pop-out window of RFID ID 584

When we view the attendance at the attendance list, we can see the owner of the RFID ID's name has been stored in it.

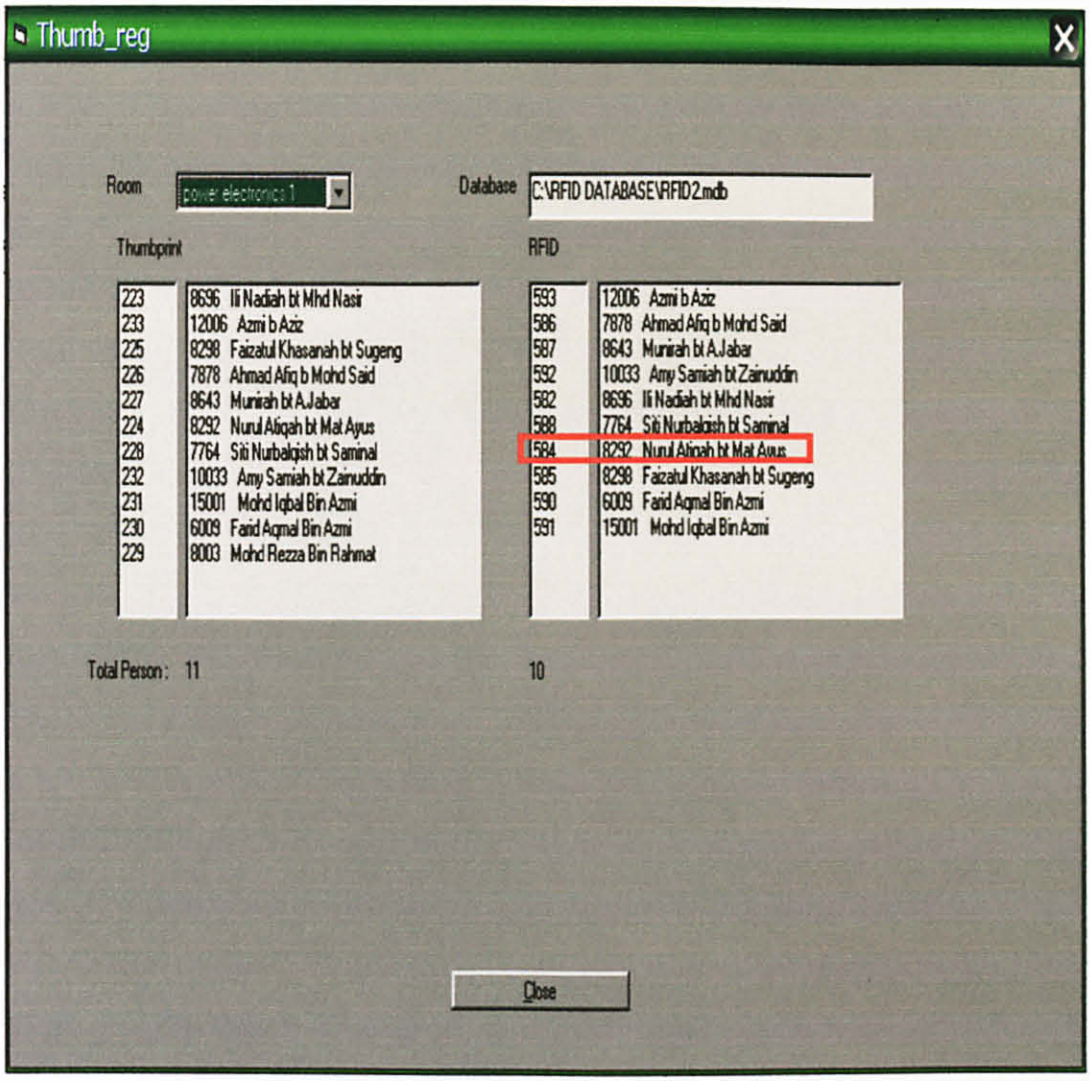

Figure 38: Name of RFID ID 584 appeared in attendance list

4.4 Limitation of the Developed System

There are several limitations have been detected while finishing this project. These limitations need to be observed more and overcomes need to be done.

The limitations are:

- Active RFID tags consume much energy from the batteries; therefore life time of the batteries is short.
- The reader can only detect the tags at certain ranges.
- Accurate read rates on some tags are very low due to battery and interference problems.
- Physical limitations like reading through metals still exist even though not very influential.
- The students can still pass their cards to their friends that join the lecture after they scan their thumbprints

CHAPTER 5
CONCLUSION AND RECOMMENDATION

5.1 Conclusion

As conclusion, the objective of this project has been successfully achieved. The coding has been successfully developed to apply thumbprint scanning for attendance identification purposes. The system is reliable and can be used in any organization for more dependable attendance record. Develop the coding and the database were the biggest challenge during finishing the project. I have learnt how to conduct the software that I am not familiar with.

5.2 Recommendation

a. Change the RS232 with API file for better connection between the readers and the database
b. Regular check of active tag's battery
c. Try to run the system by using the hardware of the thumbprint scanner
d. Find the most suitable position for reader to be located at.

REFERENCES

[1] Ron Weinstein, John Hopkins University; (vol.7 no.3); RFID: Technical Overview and ITS Application to the Enterprise; May/June 2005

[2] Simson Garfinkel, Beth Rosenberg; RFID: Application, Security and Privacy; July 2005

[3] Francisco Silva, Vitor Filipe and Antonio Pereira; Automatic Control of Students' Attendance in Classroom s Using RFID; 31st October 2008

[4] Wiley & Sons LTD; Klaus Finkenzeller; Fundamentals and Applications in Contactless Smart Cards and Identification; 2nd eddition: ISBN: 0-470-84402-7; April 2003

[5] Bill Glover, Himanshu Bhatt; Publisher: O'Reilly Media; RFID Essentials; January 2006

[6] Yahya Haghiri, Thomas Tarantino; Smart Card Manufacturing: A Practical Guide; ISBN: 978-0-471-49767-7; April 2002

[7] Dennis E. Brown; RFID Implementation; Sept. 21, 2006

[8] Sandip Lahiri; RFID Sourcebook; Sept. 10, 2005

[9] Amal Graafstra; RFID Toys: 11 Cool Projects for Home, Office and Entertainment (ExtremeTech); Feb. 13, 2006

[10] Manish Bhuptani and Shahram Moradpour; RFID Field Guide: Deploying Radio Frequency Identification Systems; Feb. 18, 2005

[11] Daniel Mark Dobkin; The RF in RFID: Passive UHF RFID in Practice; Sept. 27, 2007

[12] Ruud M. Bolle, Jonathan H. Connell, Sharath Pankanti, and Nalini K. Ratha; Guide to Biometrics; Feb. 19, 2010

[13] Davide Maltoni, Dario Maio, Anil K. Jain, and Salil Prabhakar; Handbook of Fingerprint Recognition; June 12, 2009

[14] Setlak, D.R.; ''Advances in Biometric Fingerprint Technology are Driving Rapid Adoption in Consumer Marketplace.''; December 13, 2005

[15] Daniel M. Dobkin; The RF in RFID: physical layer operation of passive UHF tags and readers; October 2005

APPENDIXES

APPENDIX A
(Coding for RFID MAIN MENU; cls room module)

```
Dim WithEvents adoPrimaryRS As Recordset
Private DoingRequery As Boolean
Public Event MoveComplete()

Private Sub Class_Initialize()
  Dim db As Connection
  Set db = New Connection
  db.CursorLocation = adUseClient
  db.Open "PROVIDER=Microsoft.Jet.OLEDB.3.51;Data Source=" &
MainMenu.dbname.Text
  Set adoPrimaryRS = New Recordset
  adoPrimaryRS.Open "select [Room ID],[Room Name] from Room", db,
adOpenStatic, adLockOptimistic

  DataMembers.Add "Primary"
End Sub

Private Sub Class_GetDataMember(DataMember As String, Data As
Object)
  Select Case DataMember
  Case "Primary"
    Set Data = adoPrimaryRS
  End Select
End Sub

Private Sub adoPrimaryRS_MoveComplete(ByVal adReason As
ADODB.EventReasonEnum, ByVal pError As ADODB.Error, adStatus As
ADODB.EventStatusEnum, ByVal pRecordset As ADODB.Recordset)
  RaiseEvent MoveComplete
End Sub

Private Sub adoPrimaryRS_WillChangeRecord(ByVal adReason As
ADODB.EventReasonEnum, ByVal cRecords As Long, adStatus As
ADODB.EventStatusEnum, ByVal pRecordset As ADODB.Recordset)
  'This is where you put validation code
  'This event gets called when the following actions occur
  Dim bCancel As Boolean
  Select Case adReason
  Case adRsnAddNew
  Case adRsnClose
  Case adRsnDelete
  Case adRsnFirstChange
  Case adRsnMove
  Case adRsnRequery
  Case adRsnResynch
  Case adRsnUndoAddNew
  Case adRsnUndoDelete
  Case adRsnUndoUpdate
  Case adRsnUpdate
  End Select

  If bCancel Then adStatus = adStatusCancel
End Sub

Public Property Get EditingRecord() As Boolean
```

```
  EditingRecord = (adoPrimaryRS.EditMode <> adEditNone)
End Property

Public Property Get AbsolutePosition() As Long
  AbsolutePosition = adoPrimaryRS.AbsolutePosition
End Property

Public Sub AddNew()
  adoPrimaryRS.AddNew
End Sub

Public Sub Delete()
  adoPrimaryRS.Delete
    MoveNext
End Sub

Public Sub Requery()
  adoPrimaryRS.Requery
  DataMemberChanged "Primary"
End Sub

Public Sub Update()
  With adoPrimaryRS
    .UpdateBatch adAffectAll
    If .EditMode = adEditAdd Then
      MoveLast
    End If
  End With
End Sub

Public Sub Cancel()
  With adoPrimaryRS
    .CancelUpdate
    If .EditMode = adEditAdd Then
      MoveFirst
    End If
  End With
End Sub

Public Sub MoveFirst()
  adoPrimaryRS.MoveFirst
End Sub

Public Sub MoveLast()
  adoPrimaryRS.MoveLast
End Sub

Public Sub MoveNext()
  If Not adoPrimaryRS.EOF Then adoPrimaryRS.MoveNext
  If adoPrimaryRS.EOF And adoPrimaryRS.RecordCount > 0 Then
    Beep
    'moved off the end so go back
    adoPrimaryRS.MoveLast
  End If
End Sub
```

```
Public Sub MovePrevious()
  If Not adoPrimaryRS.BOF Then adoPrimaryRS.MovePrevious
  If adoPrimaryRS.BOF And adoPrimaryRS.RecordCount > 0 Then
    Beep
    'moved off the end so go back
    adoPrimaryRS.MoveFirst
  End If
End Sub
```

APPENDIX B
(Coding for RFID MAIN MENU; cls student module)

```
Dim WithEvents adoPrimaryRS As Recordset
Private DoingRequery As Boolean
Public Event MoveComplete()

Private Sub Class_Initialize()
  Dim db As Connection
  Set db = New Connection
  db.CursorLocation = adUseClient
  db.Open "PROVIDER=Microsoft.Jet.OLEDB.3.51;Data Source=" &
MainMenu.dbname.Text



  Set adoPrimaryRS = New Recordset
  adoPrimaryRS.Open "select
ID,Student_ID,Thumbprint,RFID,Name,MobilePhone,EmailAddress from
Student", db, adOpenStatic, adLockOptimistic

  DataMembers.Add "Primary"
End Sub

Private Sub Class_GetDataMember(DataMember As String, Data As
Object)
  Select Case DataMember
  Case "Primary"
    Set Data = adoPrimaryRS
  End Select
End Sub

Private Sub adoPrimaryRS_MoveComplete(ByVal adReason As
ADODB.EventReasonEnum, ByVal pError As ADODB.Error, adStatus As
ADODB.EventStatusEnum, ByVal pRecordset As ADODB.Recordset)
  RaiseEvent MoveComplete
End Sub

Private Sub adoPrimaryRS_WillChangeRecord(ByVal adReason As
ADODB.EventReasonEnum, ByVal cRecords As Long, adStatus As
ADODB.EventStatusEnum, ByVal pRecordset As ADODB.Recordset)
  'This is where you put validation code
  'This event gets called when the following actions occur
  Dim bCancel As Boolean
  Select Case adReason
  Case adRsnAddNew
  Case adRsnClose
  Case adRsnDelete
  Case adRsnFirstChange
  Case adRsnMove
  Case adRsnRequery
  Case adRsnResynch
  Case adRsnUndoAddNew
  Case adRsnUndoDelete
  Case adRsnUndoUpdate
  Case adRsnUpdate
  End Select
```

```
  If bCancel Then adStatus = adStatusCancel
End Sub

Public Property Get EditingRecord() As Boolean
  EditingRecord = (adoPrimaryRS.EditMode <> adEditNone)
End Property

Public Property Get AbsolutePosition() As Long
  AbsolutePosition = adoPrimaryRS.AbsolutePosition
End Property

Public Sub AddNew()
  adoPrimaryRS.AddNew
End Sub

Public Sub Delete()
  adoPrimaryRS.Delete
    MoveNext
End Sub

Public Sub Requery()
  adoPrimaryRS.Requery
  DataMemberChanged "Primary"
End Sub

Public Sub Update()
  With adoPrimaryRS
    .UpdateBatch adAffectAll
    If .EditMode = adEditAdd Then
      MoveLast
    End If
  End With
End Sub

Public Sub Cancel()
  With adoPrimaryRS
    .CancelUpdate
    If .EditMode = adEditAdd Then
      MoveFirst
    End If
  End With
End Sub

Public Sub MoveFirst()
  adoPrimaryRS.MoveFirst
End Sub

Public Sub MoveLast()
  adoPrimaryRS.MoveLast
End Sub

Public Sub MoveNext()
  If Not adoPrimaryRS.EOF Then adoPrimaryRS.MoveNext
  If adoPrimaryRS.EOF And adoPrimaryRS.RecordCount > 0 Then
    Beep
```

```
    'moved off the end so go back
    adoPrimaryRS.MoveLast
  End If
End Sub

Public Sub MovePrevious()
  If Not adoPrimaryRS.BOF Then adoPrimaryRS.MovePrevious
  If adoPrimaryRS.BOF And adoPrimaryRS.RecordCount > 0 Then
    Beep
    'moved off the end so go back
    adoPrimaryRS.MoveFirst
  End If
End Sub
```

APPENDIX C
(Coding for RFID MAIN MENU; Main menu)

```
Private Sub Command1_Click()
frmThumb_reg.Show

End Sub

Private Sub Command2_Click()
frmStudent1.Show
End Sub

Private Sub Command3_Click()
frmRoom1.Show
End Sub
Private Sub Command5_Click()
Close
End
End Sub

Private Sub Config_Click()
frmConfig.Text1 = dbname
frmConfig.Show
End Sub

Private Sub Form_Activate()
Dim mydb
On Error Resume Next
Open "c:\rfid.ini" For Input As #1
If Err Then
MsgBox ("config file not found... Please set your database...")
frmConfig.Show
Close
Exit Sub

End If

Input #1, mydb
dbname = mydb
Close
On Error Resume Next
frmThumb_reg.Data1.DatabaseName = dbname
frmThumb_reg.Data2.DatabaseName = dbname
frmThumb_reg.Data3.DatabaseName = dbname
frmThumb_reg.Data4.DatabaseName = dbname
frmThumb_reg.Data5.DatabaseName = dbname



End Sub
```

APPENDIX D
(Coding for RFID MAIN MENU; database)

```
Private Sub cmdClose_Click()
  Unload Me
End Sub

Private Sub DBCombo1_Change()
List1.Clear
List2.Clear
List3.Clear
List4.Clear
'Text1 = DBCombo1.Text
Data3.Recordset.MoveFirst

Data3.Recordset.FindFirst "[Room Name] like '" & DBCombo1.Text &
"'":

 If Data3.Recordset.NoMatch = False Then
 Text1 = Data3.Recordset("Room ID")
 Exit Sub
Data3.Recordset.MoveFirst
End If


Do While Not Data3.Recordset.EOF


Data3.Recordset.FindNext "[Room Name] like '" & DBCombo1.Text & "'":


 If Data3.Recordset.NoMatch = False Then
 On Error Resume Next

  Text1 = Data3.Recordset("Room ID")

 Else

 Exit Do
End If

Loop

'*********************************************


End Sub
```

```
Private Sub Form_Load()
Data1.DatabaseName = MainMenu.dbname
Data2.DatabaseName = MainMenu.dbname
Data3.DatabaseName = MainMenu.dbname
Data4.DatabaseName = MainMenu.dbname
Data5.DatabaseName = MainMenu.dbname
Text3 = MainMenu.dbname
End Sub

Private Sub Text1_Change()
Dim TotalPerson
TotalPerson = 0

List1.Clear

'***************** Thunbprint *********************
Data2.Recordset.MoveFirst
Data2.Recordset.FindFirst "[Room ID] like '" & Text1 & "'":

 If Data2.Recordset.NoMatch = False Then

 List1.AddItem Data2.Recordset("Thumbprint ID")
 TotalPerson = TotalPerson + 1
 '############################

             List2.Clear

            Data4.Recordset.MoveFirst
            Data4.Recordset.FindFirst "[Thumbprint] like '" &
Data2.Recordset("Thumbprint ID") & "'":

             If Data4.Recordset.NoMatch = False Then
            List2.AddItem Data4.Recordset("Student_ID") & "    " &
Data4.Recordset("Name")
             'List2.AddItem Data4.Recordset("First Name")
             Else
                Data4.Recordset.MoveFirst
            End If
 '

 '#########################
Else
Data2.Recordset.MoveFirst
End If

'*****************************************************


Do While Not Data2.Recordset.EOF


Data2.Recordset.FindNext "[Room ID] like '" & Text1 & "'":
```

```
 If Data2.Recordset.NoMatch = False Then
 On Error Resume Next

 List1.AddItem Data2.Recordset("Thumbprint ID")
 TotalPerson = TotalPerson + 1
 '##############################
             Data4.Recordset.MoveFirst
             Do While Not Data4.Recordset.EOF


            Data4.Recordset.FindNext "[Thumbprint] like '" &
Data2.Recordset("Thumbprint ID") & "'":


             If Data4.Recordset.NoMatch = False Then
             On Error Resume Next

             List2.AddItem Data4.Recordset("Student_ID") & "    " &
Data4.Recordset("Name")
            Exit Do
             Else

            List2.AddItem "unknown"
            Data4.Recordset.MoveFirst
            Exit Do
            End If

            Loop




  '##############################




 Else
Exit Do

End If

Loop
Label3 = TotalPerson

Text2 = Text1
End Sub

Private Sub Text2_Change()
'Exit Sub
Dim TotalPerson
TotalPerson = 0
```

```
List4.Clear


'***************** Thunbprint ********************
Data5.Recordset.MoveFirst
Data5.Recordset.FindFirst "[Room_ID] like '" & Text2 & "'":

 If Data5.Recordset.NoMatch = False Then

 List4.AddItem Data5.Recordset("Rfid_ID")
 TotalPerson = TotalPerson + 1
 '##########################


            List3.Clear
            Data4.Recordset.MoveFirst
            Data4.Recordset.FindFirst "[RFID] like '" &
Data5.Recordset("Rfid_ID") & "'":

             If Data4.Recordset.NoMatch = False Then
            List3.AddItem Data4.Recordset("Student_ID") & "    " &
Data4.Recordset("Name")
             'List2.AddItem Data4.Recordset("First Name")
             Else
                Data4.Recordset.MoveFirst
            End If
 '

 '##########################
Else
Data2.Recordset.MoveFirst
End If

'*****************************************************


Do While Not Data5.Recordset.EOF


Data5.Recordset.FindNext "[Room_ID] like '" & Text2 & "'":


 If Data5.Recordset.NoMatch = False Then
 On Error Resume Next

 List4.AddItem Data5.Recordset("Rfid_ID")
 TotalPerson = TotalPerson + 1
 '##############################
              Data4.Recordset.MoveFirst
             Do While Not Data4.Recordset.EOF
```

```
            Data4.Recordset.FindNext "[RFID] like '" &
Data5.Recordset("Rfid_ID") & "'":


             If Data4.Recordset.NoMatch = False Then
             On Error Resume Next

             List3.AddItem Data4.Recordset("Student_ID") & "    " &
Data4.Recordset("Name")
            Exit Do
             Else

            List3.AddItem "unknown"

            Exit Do
            End If

            Loop




 '##############################




 Else
Exit Do

End If

Loop
Label5 = TotalPerson

Text2 = Text1
End Sub
```

APPENDIX E
(Coding for RFID MAIN MENU; Edit student list)

```
Private WithEvents PrimaryCLS As clsStudent
Dim mbChangedByCode As Boolean
Dim mvBookMark As Variant
Dim mbEditFlag As Boolean
Dim mbAddNewFlag As Boolean
Dim mbDataChanged As Boolean

Private Sub Form_Load()
  Set PrimaryCLS = New clsStudent

  grdDataGrid.DataMember = "Primary"
  Set grdDataGrid.DataSource = PrimaryCLS
End Sub

Private Sub Form_Resize()
  On Error Resume Next
  'This will resize the grid when the form is resized
  grdDataGrid.Height = Me.ScaleHeight - 30 - picButtons.Height -
picStatBox.Height
  lblStatus.Width = Me.Width - 1500
  cmdNext.Left = lblStatus.Width + 700
  cmdLast.Left = cmdNext.Left + 340
End Sub

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
  If mbEditFlag Or mbAddNewFlag Then Exit Sub

  Select Case KeyCode
    Case vbKeyEscape
      cmdClose_Click
    Case vbKeyEnd
      cmdLast_Click
    Case vbKeyHome
      cmdFirst_Click
    Case vbKeyUp, vbKeyPageUp
      If Shift = vbCtrlMask Then
        cmdFirst_Click
      Else
        cmdPrevious_Click
      End If
    Case vbKeyDown, vbKeyPageDown
      If Shift = vbCtrlMask Then
        cmdLast_Click
      Else
        cmdNext_Click
      End If
  End Select
End Sub

Private Sub Form_Unload(Cancel As Integer)
  Screen.MousePointer = vbDefault
End Sub

Private Sub grdDataGrid_Click()
```

```
End Sub

Private Sub PrimaryCLS_MoveComplete()
  'This will display the current record position for this recordset
  lblStatus.Caption = "Record: " & CStr(PrimaryCLS.AbsolutePosition)
End Sub

Private Sub cmdAdd_Click()
  On Error GoTo AddErr
  PrimaryCLS.MoveLast
  PrimaryCLS.AddNew
  grdDataGrid.SetFocus

  Exit Sub
AddErr:
  MsgBox Err.Description
End Sub

Private Sub cmdDelete_Click()
  On Error GoTo DeleteErr
  PrimaryCLS.Delete
  Exit Sub
DeleteErr:
  MsgBox Err.Description
End Sub

Private Sub cmdRefresh_Click()
  'This is only needed for multi user apps
  On Error GoTo RefreshErr
  PrimaryCLS.Requery
  Exit Sub
RefreshErr:
  MsgBox Err.Description
End Sub

Private Sub cmdEdit_Click()
  On Error GoTo EditErr

  lblStatus.Caption = "Edit record"
  mbEditFlag = True
  SetButtons False
  Exit Sub

EditErr:
  MsgBox Err.Description
End Sub
Private Sub cmdCancel_Click()
  On Error Resume Next

  PrimaryCLS.Cancel
  SetButtons True
End Sub

Private Sub cmdUpdate_Click()
  On Error GoTo UpdateErr
```

```
    PrimaryCLS.Update
    SetButtons True
    Exit Sub
UpdateErr:
    MsgBox Err.Description
End Sub

Private Sub cmdClose_Click()
    Unload Me
End Sub

Private Sub cmdFirst_Click()
    On Error GoTo GoFirstError

    PrimaryCLS.MoveFirst
    mbDataChanged = False

    Exit Sub

GoFirstError:
    MsgBox Err.Description
End Sub

Private Sub cmdLast_Click()
    On Error GoTo GoLastError

    PrimaryCLS.MoveLast
    mbDataChanged = False

    Exit Sub

GoLastError:
    MsgBox Err.Description
End Sub

Private Sub cmdNext_Click()
    On Error GoTo GoNextError

    PrimaryCLS.MoveNext
    Exit Sub
GoNextError:
    MsgBox Err.Description
End Sub

Private Sub cmdPrevious_Click()
    On Error GoTo GoPrevError

    PrimaryCLS.MovePrevious
    Exit Sub

GoPrevError:
    MsgBox Err.Description
End Sub
```

```
Private Sub SetButtons(bVal As Boolean)
  cmdAdd.Visible = bVal
  cmdUpdate.Visible = Not bVal
  cmdCancel.Visible = Not bVal
  cmdDelete.Visible = bVal
  cmdClose.Visible = bVal
  cmdRefresh.Visible = bVal
  cmdNext.Enabled = bVal
  cmdFirst.Enabled = bVal
  cmdLast.Enabled = bVal
  cmdPrevious.Enabled = bVal
End Sub
```

APPENDIX F
(Coding for RFID MAIN MENU; Edit room list)

```
Private WithEvents PrimaryCLS As clsRoom
Dim mbChangedByCode As Boolean
Dim mvBookMark As Variant
Dim mbEditFlag As Boolean
Dim mbAddNewFlag As Boolean
Dim mbDataChanged As Boolean

Private Sub Form_Load()
  Set PrimaryCLS = New clsRoom

  grdDataGrid.DataMember = "Primary"
  Set grdDataGrid.DataSource = PrimaryCLS
End Sub

Private Sub Form_Resize()
  On Error Resume Next
  'This will resize the grid when the form is resized
  grdDataGrid.Height = Me.ScaleHeight - 30 - picButtons.Height -
picStatBox.Height
  lblStatus.Width = Me.Width - 1500
  cmdNext.Left = lblStatus.Width + 700
  cmdLast.Left = cmdNext.Left + 340
End Sub

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
  If mbEditFlag Or mbAddNewFlag Then Exit Sub

  Select Case KeyCode
    Case vbKeyEscape
      cmdClose_Click
    Case vbKeyEnd
      cmdLast_Click
    Case vbKeyHome
      cmdFirst_Click
    Case vbKeyUp, vbKeyPageUp
      If Shift = vbCtrlMask Then
        cmdFirst_Click
      Else
        cmdPrevious_Click
      End If
    Case vbKeyDown, vbKeyPageDown
      If Shift = vbCtrlMask Then
        cmdLast_Click
      Else
        cmdNext_Click
      End If
  End Select
End Sub

Private Sub Form_Unload(Cancel As Integer)
  Screen.MousePointer = vbDefault
End Sub

Private Sub grdDataGrid_Click()
```

```
End Sub

Private Sub PrimaryCLS_MoveComplete()
  'This will display the current record position for this recordset
  lblStatus.Caption = "Record: " & CStr(PrimaryCLS.AbsolutePosition)
End Sub

Private Sub cmdAdd_Click()
  On Error GoTo AddErr
  PrimaryCLS.MoveLast
  PrimaryCLS.AddNew
  grdDataGrid.SetFocus

  Exit Sub
AddErr:
  MsgBox Err.Description
End Sub

Private Sub cmdDelete_Click()
  On Error GoTo DeleteErr
  PrimaryCLS.Delete
  Exit Sub
DeleteErr:
  MsgBox Err.Description
End Sub

Private Sub cmdRefresh_Click()
  'This is only needed for multi user apps
  On Error GoTo RefreshErr
  PrimaryCLS.Requery
  Exit Sub
RefreshErr:
  MsgBox Err.Description
End Sub

Private Sub cmdEdit_Click()
  On Error GoTo EditErr

  lblStatus.Caption = "Edit record"
  mbEditFlag = True
  SetButtons False
  Exit Sub

EditErr:
  MsgBox Err.Description
End Sub
Private Sub cmdCancel_Click()
  On Error Resume Next

  PrimaryCLS.Cancel
  SetButtons True
End Sub

Private Sub cmdUpdate_Click()
  On Error GoTo UpdateErr
```

```
    PrimaryCLS.Update
    SetButtons True
    Exit Sub
UpdateErr:
    MsgBox Err.Description
End Sub

Private Sub cmdClose_Click()
    Unload Me
End Sub

Private Sub cmdFirst_Click()
    On Error GoTo GoFirstError

    PrimaryCLS.MoveFirst
    mbDataChanged = False

    Exit Sub

GoFirstError:
    MsgBox Err.Description
End Sub

Private Sub cmdLast_Click()
    On Error GoTo GoLastError

    PrimaryCLS.MoveLast
    mbDataChanged = False

    Exit Sub

GoLastError:
    MsgBox Err.Description
End Sub

Private Sub cmdNext_Click()
    On Error GoTo GoNextError

    PrimaryCLS.MoveNext
    Exit Sub
GoNextError:
    MsgBox Err.Description
End Sub

Private Sub cmdPrevious_Click()
    On Error GoTo GoPrevError

    PrimaryCLS.MovePrevious
    Exit Sub

GoPrevError:
    MsgBox Err.Description
End Sub
```

```
Private Sub SetButtons(bVal As Boolean)
  cmdAdd.Visible = bVal
  cmdUpdate.Visible = Not bVal
  cmdCancel.Visible = Not bVal
  cmdDelete.Visible = bVal
  cmdClose.Visible = bVal
  cmdRefresh.Visible = bVal
  cmdNext.Enabled = bVal
  cmdFirst.Enabled = bVal
  cmdLast.Enabled = bVal
  cmdPrevious.Enabled = bVal
End Sub
```

APPENDIX G
(Coding for RFID Simulator; Main Menu)

```
Private Sub Command1_Click()
frmRfid_reg.Show

End Sub

Private Sub Command2_Click()
Text1 = ""
Text2 = ""
Text3 = ""
Text5 = ""
End Sub

Private Sub Command3_Click()
frmRoom.Show
End Sub
Private Sub Command5_Click()
Close
End
End Sub

Private Sub Config_Click()
frmConfig.Text1 = dbname
frmConfig.Show
End Sub

Private Sub Form_Activate()
Dim mydb
On Error Resume Next
Open "c:\rfid.ini" For Input As #1
If Err Then
MsgBox ("config file not found... Please set your database...")
frmConfig.Show
Close
Exit Sub

End If

Input #1, mydb
dbname = mydb
Close

End Sub

Private Sub Form_Load()



' Buffer to hold input string
   Dim Instring As String
   ' Use COM1.
   MSComm1.CommPort = 1
   ' 9600 baud, no parity, 8 data, and 1 stop bit.
   MSComm1.Settings = "115200,N,8,1"
   ' Tell the control to read entire buffer when Input
   ' is used.
```

```
    MSComm1.InputLen = 0
    ' Open the port.
    MSComm1.PortOpen = True
  ' Do
   '    DoEvents
    ' Buffer$ = Buffer$ & MSComm1.Input
    'tc = Chr$(11) & Chr$(3)
    'Loop Until InStr(Buffer$, tc & vbCrLf)
    ' Text1 = 584


    ' Read the "OK" response data in the serial port.
    ' Close the serial port.

    ' Read the "OK" response data in the serial port.
    ' Close the serial port.
    'MSComm1.PortOpen = False


End Sub

Private Sub Text1_Change()
Dim r
Dim tc(2000) As String



For r = 1 To (Len(Text1))

'Do While Len(Text1)
'r = r + 1
tc(r) = Mid$(Text1, r, 1)

'If tc(r) <> "" Then
Text2 = Text2 + Str(Asc(tc(r)))
'End If
'Text2 = Len(Text1)
'Loop
Next
If Mid(Text2, 2, 15) = "126 49 9 17 3 1" Then
'Text3 = Mid(Text1, 1, 5) 'Chr(126) & Chr(49) & Chr(9) & Chr(17) &
Chr(3) & Chr(1)
Text3 = "3"
Text5 = "584"
End If
'Text3 = Mid(Text2, 2, 15)
'Text3 = Chr(126) & Chr(49) & Chr(9) & Chr(17) & Chr(3) & Chr(1)
End Sub

Private Sub Text5_Change()
If Text5 = "" Then Exit Sub
Data2.Recordset.MoveFirst
' find first
Data2.Recordset.FindFirst "[Rfid_ID] like '" & Text5 & "'":
```

```
 If Data2.Recordset.NoMatch = False Then
   Data2.Recordset.Edit
   Data2.Recordset("Room_ID") = Text3
    MsgBox "Rfid_ID " & Text5 & " was edit in RFID Database"
    Data2.Recordset.Update
    Exit Sub
End If

'find next

Do While Not Data2.Recordset.EOF
Data2.Recordset.MoveFirst

Data2.Recordset.FindNext "[Rfid_ID] like '" & Text5 & "'":


 If Data2.Recordset.NoMatch = False Then
 On Error Resume Next
  'Edit Existing
  Data2.Recordset.Edit
  Data2.Recordset("Room_ID") = Text3
   MsgBox "Thumbprint ID " & Text5 & " was edit in RFID Database"
   Data2.Recordset.Update
  Exit Do
 Else
'add new
Data2.Recordset.AddNew
Data2.Recordset("Room_ID") = Text3
Data2.Recordset("Rfid_ID") = Text5
MsgBox "Rfid_ID " & Text5 & " was add to RFID Database"
 Exit Do
End If

Loop

Data2.Recordset.Update
End Sub

Private Sub Timer1_Timer()


Text1 = Text1 + MSComm1.Input



End Sub
```

APPENDIX H
(Coding for RFID Simulator; RFID registration)

```
Private Sub cmdClose_Click()
  Unload Me
End Sub



Private Sub Command1_Click()
Data2.Recordset.MoveFirst

Data2.Recordset.FindFirst "[Rfid_ID] like '" & Text2 & "'":

 If Data2.Recordset.NoMatch = False Then
  If Data2.Recordset("Room_ID") = Text1 Then Data2.Recordset.Delete
  MsgBox "Rfid_ID " & Text2 & " was remove from RFID Database"
End If
Data2.Recordset.MoveFirst
Do While Not Data2.Recordset.EOF


Data2.Recordset.FindNext "[Rfid_ID] like '" & Text2 & "'":


 If Data2.Recordset.NoMatch = False Then
 On Error Resume Next

  If Data2.Recordset("Room_ID") = Text1 Then Data2.Recordset.Delete
   MsgBox "Rfid_ID " & Text2 & " was remove from RFID Database"
  Exit Do
 Else

 Exit Do
End If

Loop



End Sub

Private Sub DBCombo1_Click(Area As Integer)

Data3.Recordset.MoveFirst

Data3.Recordset.FindFirst "[Room Name] like '" & DBCombo1.Text &
"'":

 If Data3.Recordset.NoMatch = False Then
 Text1 = Data3.Recordset("Room ID")
 Exit Sub
Data3.Recordset.MoveFirst
End If
```

```
Do While Not Data3.Recordset.EOF

Data3.Recordset.FindNext "[Room Name] like '" & DBCombo1.Text & "'":

 If Data3.Recordset.NoMatch = False Then
 On Error Resume Next

  Text1 = Data3.Recordset("Room ID")

 Else

 Exit Do
End If

Loop

'************************************************

End Sub

Private Sub Form_Load()

End Sub

Private Sub Thumb_in_Click()
Data2.Recordset.MoveFirst

Data2.Recordset.FindFirst "[Rfid_ID] like '" & Text2 & "'":

 If Data2.Recordset.NoMatch = False Then
   Data2.Recordset.Edit
   Data2.Recordset("Room_ID") = Text1
    MsgBox "Rfid_ID " & Text2 & " was edit in RFID Database"
    Data2.Recordset.Update
    Exit Sub
End If

Do While Not Data2.Recordset.EOF

Data2.Recordset.FindFirst "[Rfid_ID] like '" & Text2 & "'":

 If Data2.Recordset.NoMatch = False Then
 On Error Resume Next
  Data2.Recordset.Edit
  Data2.Recordset("Room_ID") = Text1
   MsgBox "Rfid_ID " & Text2 & " was edit in RFID Database"
   Data2.Recordset.Update
  Exit Do
 Else
Data2.Recordset.AddNew
```

```
Data2.Recordset("Room_ID") = Text1
Data2.Recordset("Rfid_ID") = Text2
MsgBox "Rfid_ID " & Text2 & " was add to RFID Database"
 Exit Do
End If

Loop

Data2.Recordset.Update
End Sub
```

APPENDIX I
(Coding for Thumbprint Simulator; Main Menu)

```
Private Sub Command1_Click()
frmThumb_reg.Show

End Sub

Private Sub Command2_Click()
frmStudent.Show
End Sub

Private Sub Command3_Click()
frmRoom.Show
End Sub
Private Sub Command5_Click()
Close
End
End Sub

Private Sub Config_Click()
frmConfig.Text1 = dbname
frmConfig.Show
End Sub

Private Sub Form_Load()
Dim mydb
On Error Resume Next
Open "c:\rfid.ini" For Input As #1
If Err Then
MsgBox ("config file not found... Please set your database...")
frmConfig.Show
Close
Exit Sub

End If

Input #1, mydb
dbname = mydb
Close


End Sub
```

APPENDIX J

(Coding for Thumbprint Simulator; Thumbprint registration)

```
Private Sub cmdClose_Click()
  Unload Me
End Sub



Private Sub Command1_Click()
Data2.Recordset.MoveFirst

Data2.Recordset.FindFirst "[Thumbprint ID] like '" & Text2 & "'":

 If Data2.Recordset.NoMatch = False Then
  If Data2.Recordset("Room ID") = Text1 Then Data2.Recordset.Delete
  MsgBox "Thumbprint ID " & Text2 & " was remove from Thumbprint
Database"
End If
Data2.Recordset.MoveFirst
Do While Not Data2.Recordset.EOF


Data2.Recordset.FindNext "[Thumbprint ID] like '" & Text2 & "'":


 If Data2.Recordset.NoMatch = False Then
 On Error Resume Next

  If Data2.Recordset("Room ID") = Text1 Then Data2.Recordset.Delete
   MsgBox "Thumbprint ID " & Text2 & " was remove from Thumbprint
Database"
  Exit Do
 Else

 Exit Do
End If

Loop



End Sub

Private Sub DBCombo1_Click(Area As Integer)
Data3.Recordset.MoveFirst

Data3.Recordset.FindFirst "[Room Name] like '" & DBCombo1.Text &
"'":

 If Data3.Recordset.NoMatch = False Then
 Text1 = Data3.Recordset("Room ID")
 Exit Sub
Data3.Recordset.MoveFirst
End If
```

```
Do While Not Data3.Recordset.EOF

Data3.Recordset.FindNext "[Room Name] like '" & DBCombo1.Text & "'":

 If Data3.Recordset.NoMatch = False Then
 On Error Resume Next

  Text1 = Data3.Recordset("Room ID")

 Else

 Exit Do
End If

Loop

'*********************************************

End Sub

Private Sub Form_Load()

End Sub

Private Sub Thumb_in_Click()
Data2.Recordset.MoveFirst

Data2.Recordset.FindFirst "[Thumbprint ID] like '" & Text2 & "'":

 If Data2.Recordset.NoMatch = False Then
   Data2.Recordset.Edit
   Data2.Recordset("Room ID") = Text1
    MsgBox "Thumbprint ID " & Text2 & " was edit in Thumbprint
Database"
    Data2.Recordset.Update
    Exit Sub
End If
Data2.Recordset.MoveFirst
Do While Not Data2.Recordset.EOF

Data2.Recordset.FindNext "[Thumbprint ID] like '" & Text2 & "'":

 If Data2.Recordset.NoMatch = False Then
 On Error Resume Next
  Data2.Recordset.Edit
  Data2.Recordset("Room ID") = Text1
   MsgBox "Thumbprint ID " & Text2 & " was edit in Thumbprint
Database"
```

```
   Data2.Recordset.Update
  Exit Do
 Else
Data2.Recordset.AddNew
Data2.Recordset("Room ID") = Text1
Data2.Recordset("Thumbprint ID") = Text2
MsgBox "Thumbprint ID " & Text2 & " was add to Thumbprint Database"
 Exit Do
End If

Loop

Data2.Recordset.Update
End Sub
```

APPENDIX K
(Coding for Configuration)

```
Private Sub Command1_Click()
CommonDialog1.ShowOpen
If CommonDialog1.FileName <> "" Then Text1 = CommonDialog1.FileName
End Sub

Private Sub Command2_Click()
If Text1 = "" Then
MsgBox "please select your database.."
Exit Sub
End If

Open "c:\rfid.ini" For Output As #1
Write #1, Text1
Close
MainMenu.dbname = Text1
Unload Me

End Sub

Private Sub Form_Load()

End Sub
```

APPENDIX L
(Specifications of the active tag)

User Memory	0 – 256 Kbits	
Multi-Tag Read Capability	Yes	
Transmit Frequency	916 MHz, 927 MHz, or 868 MHz	
Receive Frequency	433 MHz	
Read Range	Receive	85m (280 feet)*
	Transmit	152m (500 feet)*
Power	3V Lithium-ion watch battery	
Battery Life	1 – 3 years depending on use (tag has low battery detection)	
Dimensions	85.0 mm x 54.3 mm x 5.6 mm (3.3 in x 2.1 in x 0.2 in)	
Weight	23 grams (0.8 oz)	
Case Material	ABS (Acrylonitrile Butadiene Styrene)	
Temperature	Operating	-35C to +50C (-31F to +122F)
	Storage	-40C to +85C (-40F to +185F)
Options	Memory	0 – 256Kbit memory sizes available in 2x increments

APPENDIX M
(Specification of the reader)

Functionality	Reads and writes RFID tags	
Multi-Tag Read Capability	Yes	
Transmit Frequency to Tag	433 MHz	
Receive Frequency from Tag	916 MHz, 927 MHz, or 868 MHz	
Range	30m (100 feet) to tag	
	85m (280 feet) from tag	
Host Communications	RS232	9600 - 115200 Baud
	Ethernet	10/100 Mbps
	WLAN (optional)	2.4 GHz, 5.2 GHz
Power	12Vdc, 1.5A	
Dimensions	without antennas	150 mm x 85 mm x 27 mm (5.9 in x 3.3 in x 1.1 in)
	with antennas	150 mm x 85 mm x 167 mm (5.9 in x 3.3 in x 6.6 in)
Weight	680 grams (1.5 lbs)	
Case Material	Impact resistant polystyrene with UL94-HB flammability rating	
Temperature	Operating	-35C to +50C (-31F to +122F)
	Storage	-40C to +85C (-40F to +185F)
Indicators	RF LED	On while receiving packet from tag.
	HOST LED	On while sending validated tag packet to Host.
	ACCESS LED	On while transmitting packet to tag.
	POWER LED	On when Reader is powered.
Connectors	Power	12Vdc, 1.5A
	Ethernet	RJ-45 female to Host
	Motion Detector	RJ-11 male
	Host Comm.	Same RJ-11 male to Host (DB9 female to Host optional)
	Input	Two contact sense inputs
	Output	Two isolated dry contact relay outputs